Energy-efficient Localisation: GPS Duty Cycling with Radio Ranging

Raja Jurdak, Peter Corke, Alban Cotillon, Dhinesh Dharman, Chris Crossman, Guillaume Salagnac

Energy-efficient Localisation: GPS Duty Cycling with Radio Ranging

Raja Jurdak
CSIRO ICT Centre

Peter Corke
Queensland University of Technology

Alban Cotillon
INSA-Lyon

Dhinesh Dharman
CSIRO ICT Centre

Chris Crossman
CSIRO ICT Centre

Guillaume Salagnac
INSA-Lyon

Abstract GPS is a commonly used and convenient technology for determining absolute position in outdoor environments, but its high power consumption leads to rapid battery depletion in mobile devices. An obvious solution is to duty cycle the GPS module, which prolongs the device lifetime at the cost of increased position uncertainty while the GPS is off. This paper addresses the tradeoff between energy consumption and localization performance in a mobile sensor network application. The focus is on augmenting GPS location with more energy-efficient location sensors to bound position estimate uncertainty while GPS is off. Empirical GPS and radio contact data from a large-scale animal tracking deployment is used to model node mobility, radio performance and GPS. Because GPS takes a considerable, and variable, time after powering up before it delivers a good position measurement, we model the GPS behaviour through empirical measurements of two GPS modules. These models are then used to explore duty cycling strategies for maintaining position uncertainty within specified bounds. We then explore the benefits of using short-range radio contact logging alongside GPS as an energy-inexpensive means of lowering uncertainty while the GPS is off, and we propose strategies that use RSSI ranging and GPS back-offs to further reduce energy consumption. Results show that our combined strategies can cut node energy consumption by one third while still meeting application-specific positioning criteria.

1 Introduction

Location-aware applications have gained significant interest recently, due to technology advances in GPS and in embedded communication and processing. While GPS is the dominant technology for providing location information, it has a relatively high power consumption (typically in excess of 100mW) that is problematic for mobile devices with small batteries. Mobile network proposals often assume that GPS can provide absolute location information [ID03] with a given uncertainty, without considering the full range of operational constraints and the very high energy usage of this technology. The obvious solution is to duty cycle the GPS module to prolong node lifetime. This comes at the cost of increased position uncertainty whenever the GPS is powered down. Fortunately, most applications can tolerate a certain amount of localization uncertainty. If a model of the node's motion is known then we can estimate position uncertainty as a function of time.

This paper proposes the use of additional low-power sensor data to infer node locations and to reduce position uncertainty while the GPS is off. Specifically, we consider radio ranging

[0]ACM TOSN – Publication date: March 2013 (Received Dec. 2010; revised Feb. 2012; accepted Feb. 2012)

between neighbouring nodes. Nodes send radio messages to share their position and estimated uncertainty with spatially proximal neighbours, which enables sharing of GPS load among the nodes and an overall reduction in the use of GPS at each node. We also consider an environment where there is no infrastructure — no cellular towers or fixed wireless sensor nodes — due to the vast size and remoteness of the areas in which the mobile nodes must operate.

Our initial analysis reveals that the simple combination of GPS and contact logging reduces the power consumption of the GPS module but incurs additional power consumption for increased radio activity. We then focus on modeling the evolution of position uncertainty over time and the energy implications of different GPS duty cycling strategies and radio contact logging. Our models use empirical data from long term animal tracking experiments [WCH+07] where the GPS is always on. The data provides ground truth position data to build models of node mobility and GPS performance.

This paper considers two classes of positioning applications: (1) applications which can tolerate a fixed uncertainty in node positions and are representative of location-based mobile phone applications, vehicle tracking applications, and animal tracking applications; and (2) applications where the tolerance for position uncertainty varies based on position, which are representative of virtual fencing [WCH+07] and collaborative gaming applications. While we propose a strategy for this second class of applications that can more aggressively reduce energy consumption, the main focus of the paper is concerned with more common applications that belong to the first class.

The strategy of combining GPS duty cycling and radio contact logging extends our recent work [JCDS10] to include comprehensive modelling of two different GPS modules, modelling the time to achieve a user-defined acceptable uncertainty rather than just GPS lock, and algorithms for beacon forwarding and event-based beacons.

The contributions of this paper are the following:

- Empirical characterisation of the lock time models for two GPS modules that relate position accuracy to on-time and surprisingly, the preceding off-time.
- New GPS duty cycling algorithms, based on the lock time model, and their evaluation using a detailed simulation and a large empirical dataset.
- Novel methods for augmenting duty-cycled GPS with radio ranging strategies that include periodic and event-based radio beaconing, beacon forwarding and RSSI-based ranging.

The remainder of the paper is organised as follows. Section 2 presents the related work in low-power mobile localization. Section 3 introduces cooperative localisation, formally defines the problem and the performance measures, and develops probabilistic models of GPS performance and node mobility. We first explore GPS duty cycling in Section 4 and then investigate the effect of including radio proximity in Section 5. Section 6 discusses the results and concludes the paper.

2 Related Work

This section presents recent work on GPS variants to reduce time to fix, animal tracking using embedded networked systems, and energy-efficient localisation.

2.1 GPS Variants

Several variants of GPS receivers exist including Assisted GPS (AGPS) and Differential GPS (DGPS) [PPP11]. AGPS receivers are commonly used in smart phones and rely on

the cellular network and an AGPS server to provide seed data: approximate location, reference time, and ephemeris data. This enables the AGPS device to receive signals from the visible satellites and calculate its position more quickly, reducing lock time and therefore average power consumption. However, AGPS requires the availability of hardware that is capable of cellular communication, which is a limiting assumption for applications that use non-cellular devices, such as for animal tracking. In addition, tracking wildlife and cattle in remote or rural areas means that cellular signals are not always present. As a result, we do not constrain our design by an AGPS assumption.

DGPS also assumes the availability of land-based or satellite-based infrastructure to transmit correction data that reduces the time to first fix of a GPS module. As for land-based RF infrastructure, the availability of Low Earth Orbit (LEO) satellites for transmitting this correction data is still not validated in practice, and may not apply to deployments that do not have communication access to these satellites. Patil et al. [PPP11] point out that both AGPS and DGPS do not sufficiently reduce the time to first fix on a cold start of the GPS receiver. Their proposed optimisations of post processing pseudoranges at the base stations to reduce fix times are complimentary to our contributions in this paper, as they can enhance the benefits of GPS duty cycling.

2.2 Animal Tracking

An early instance of animal tracking by networked embedded systems is the Electronic Shepherd project [TSBW04], which uses GPS localization for real-time tracking of herds of sheep in the mountains. "Real-time" is obtained thanks to GPRS modems in the collar, even though position is recorded locally and sent back to base only a few times per day in order to prolong battery life. Unlike our work, nodes are non-homogeneous: only a few animals in the herd wear collars including GPS and GPRS modules. All the others wear a small ear-tag including only a microcontroller and a radio-chip. The Networked Cow project[BCPR04] used PDAs with GPS and adhoc-mode WiFi to route position information to a base station. The ZebraNet project [Zea05] reports individual position records for zebras every few minutes. In order to make the energy problem more tractable ZebraNet collars include a solar panel, which assume that the panels are resilient to normal animal activities. Positioning is done by GPS only, and the nodes propagate their information by flooding in order to facilitate data acquisition by the mobile sink.

2.3 Energy-efficient Localization

The work in [WXL04] considers energy efficient localization for base nodes on the path of the mobile nodes, and the energy implications of each approach. While the motion models in [WXL04] are similar to this paper, their work is purely simulation-based and does not consider the benefits of contact logging or online adaptation according to energy and position changes. The work in [TKAGK05] addresses the tradeoff between localization accuracy and energy efficiency. It considers static, dynamic, and dead reckoning mobility models and studies their effect on accuracy and energy through ns-2 simulations. Our work addresses the same tradeoff but uses empirical GPS data to conduct analysis on three different speed models and a realistic energy model of the GPS-enabled nodes that includes the previously unreported stochastic lock time model. We also consider the effect of dynamically changing the required localization performance based on the mobile node's distance from an exclusion zone. Furthermore, we explore contact logging as a complement to GPS for achieving a better balance between accuracy and efficiency. In a similar fashion, Pattern et al. [PPK03] provide a tuning knob to obtain various energy-accuracy tradeoffs by the careful choice of duty cycle and activation radius.

The works in [You08] and [YHC+08] address this tradeoff as well by considering improvements to the Radio Interferometric Positioning (RIP) method [KSB+07] as the baseline localization method. A key difference between You's work and ours is that we support a variable uncertainty bound. Another distinction is their use of mote-based RSSI localization schemes in indoor environments, which require a deployed network infrastructure, consume less power, and require a calibration phase. Finally, they do not consider contact logging for reducing localization uncertainty.

Recent work [CCC+06] [LZZ06] has also explored collaborative localization that enables nodes to detect neighboring signals for position refinement. Chan et al. [CCC+06] use a cluster-based approach, where nearby nodes can establish clusters through IEEE 802.15.4 radios as a neighbour detection sensor that enhances WiFi localization. Our approach is similar in its reliance on contact signals, but it differs in that the contact signal is sent and received by the same radio, eliminating the need for multi-radio nodes. Liu et al. [LZZ06] also use a form of contact logging that establishes anchor nodes that know their location (pseudo-anchor nodes) that estimate their location based on contact with anchor nodes, and free nodes that rely on anchor or pseudo-anchor nodes for their location. This resembles the multi-hop contact logging in our virtual fencing application. However, their work focuses on algorithms to bound the uncertainty propagation in iterative localization techniques through the use of static anchor nodes, while we focus on bounding the localization uncertainty of mobile anchors and neighbouring nodes that rely on them within an energy constraint. In addition, nodes dynamically become anchors in our model based on their current GPS state.

More recently, research has explored energy-aware localization for mobile phones equipped with GPS. For example Constandache *et. al* [CGea09] propose an average error metric for GPS duty cycling. Using proximity mechanisms like Wi-Fi or GSM yields a lower average error for the same energy budget as plain GPS duty cycling. One of their mobility models deduces the current location of the node from recent history when the user is driving on a straight road. Similarly, the work in [PKG10] stores historical information of locations where GPS does not work, and of average speed as a function of time and place. Our algorithm stores no neighbour history, where each node tracks only its own uncertainty through contact logging with neighbours. This minimises local storage requirements for individual nodes and training requirements for establishing historical patterns.

The use of accelerometers has also been proposed as a low power indicator of movement to supplement GPS duty cycling [PKG10][KLGT09]. Guo et al. [GPC+09] also consider the use of directional and angular speed for cattle behaviour classification. While the inclusion of inertial sensor readings, such as magnetometers or accelerometers, is an attractive extension to our approach, the study in this paper focuses on tracking devices that are not equipped with inertial sensors for localisation.

Two recent works [PKG10][LKea10] consider Bluetooth as a radio ranging technology, since it is pervasive in mobile phones. Bluetooth is relatively energy-hungry for neighbour discovery and frequent beaconing, as it relies on a synchronous connection-oriented master slave topology. Our work uses lower power IEEE 802.15.4 for radio ranging that use asynchronous broadcast channels and a flat topology. This enables a single beacon to be detected by all neighbours within radio range, providing a more scalable approach.

3 Cooperative localization

The notation used in this paper is illustrated in Figure 1. At time t_k we obtain a GPS measurement x_k and then turn off the GPS. At the next measurement x_{k+1}, the mobile node may be either inside or outside the user-set absolute acceptable uncertainty (AAU) bound, indicated by the large solid circle. Whenever x_{k+1} falls outside the larger circle, we denote this

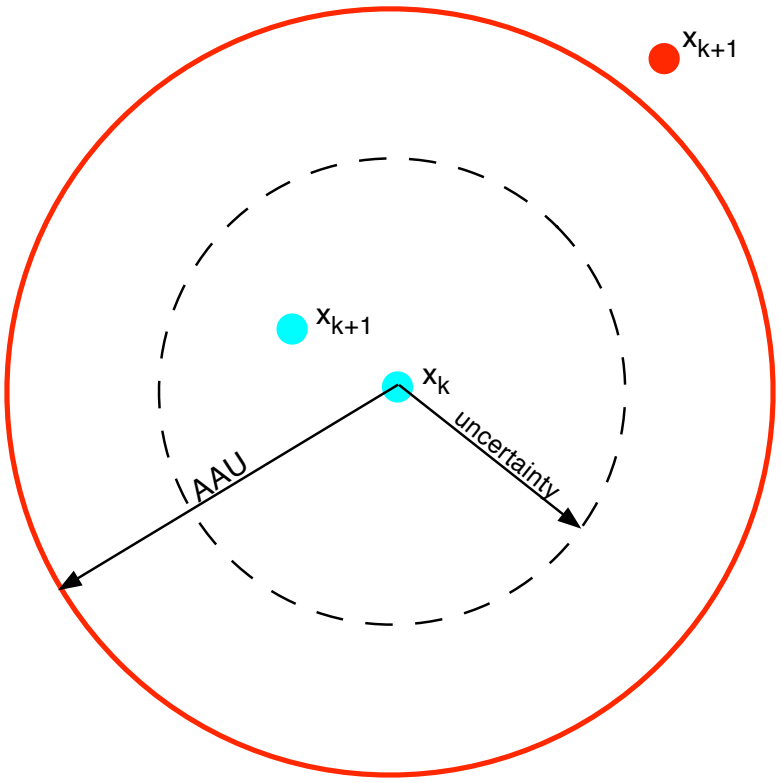


Figure 1: The growing uncertainty (dashed circle) and the maximum uncertainty bound (solid red circle).

as an *error*, since the system would have failed to deliver a position estimate within the user-set uncertainty bound. A primary performance metric is the ratio of the number of errors to the total number of GPS measurements, which we denote as the error rate σ. For an application the user sets the upper bound σ^*, which we refer to as error tolerance, and the AAU which can be considered as a precision measure. Our second performance metric is the average node power consumption, which is always in tension with the error rate σ, and is determined by the user to ensure the desired node lifetime for a specified battery configuration. It is important to note that we are not attempting to estimate the node position while the GPS is off — our aim is to estimate the uncertainty bound and to ensure that the next GPS measurement occurs within a defined radius of the previous measurement.

A key element of our approach is estimating the uncertainty bound (dashed circle in Figure 1) at any given time. Since we do not know how fast nodes are moving or in what direction, the simplest model is to assume the nodes are moving at a certain a speed and to grow the estimated uncertainty linearly with time. When uncertainty approaches the maximum allowable bound, taking into account the GPS's lock time[1] we turn on the GPS. If the node's speed exceeds our assumption, it could end up at the point outside the circle (marked in red) when the GPS obtains a fix.

3.1 Node Mobility

We consider three attributes of the node mobility model: (1) the distribution of node speed; (2) an estimate of the current speed based on the previously observed speeds; and (3) the spatial clustering behaviour of the mobile entities. We illustrate these attributes using GPS position

[1]The GPS module requires a variable time to obtain a position fix, which we denote as *lock time* and explore in Section 3.4.

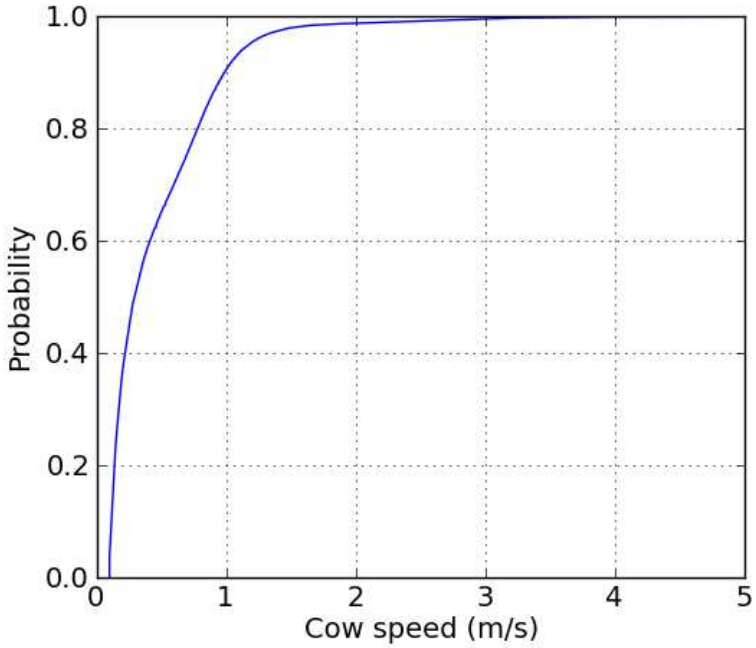


Figure 2: Cumulative distribution of observed animal speed. 95% of measurement are below 1m/s, 100% are below 3.5m/s

traces from 30 cow collar nodes collected continuously at 1 Hz over 2 days from a herd at the Belmont Research Station in central Queensland.

The distribution of node speed is useful for estimating the assumed speed of the mobile device, as well as the likelihood of variation from that average speed. Figure 2 shows the cumulative distribution function for cow speed based on the position trace. This distribution clearly shows that the cows are slow-moving animals, with a speed below 1 m/sec for 95% of the measurements. This serves as a valuable input for the assumed speed in our algorithm (see Section 4).

The speed at successive samples is not an independent random variable and is strongly persistent. Figure 3 plots the current speed as a function of speed at the previous time instant. It clearly shows a strong persistence in speed values. Consequently, we can estimate future speed predictions as a function of the currently measured speed.

Clustering is another important aspect of node mobility. The clustering behavior builds on the time series of separation distances between every pair of mobile entities. Separation distance information can be obtained using any combination of received signal strength indicator (RSSI) or the difference in exchanged GPS coordinates. The clustering behavior can be described by statistics on the separation distance between the entities and also the duration for which the entities are in contact. These statistics provide expected node density and expected contact durations among mobile nodes, which in turn guides the decision on setting GPS duty cycle and radio beaconing parameters. Figure 4 shows the pairwise internode distances for our cow dataset. The dominance of shorter internode distance drives our design decisions for radio beaconing in Section 5.

3.2 Energy Model

Our approach tracks node energy consumption by recording the duty cycles of the major node components [JRO08]: GPS module, radio, microcontroller unit (MCU), and other components.

GPS duty cycling is a key factor in our approach. We simply compute the GPS module

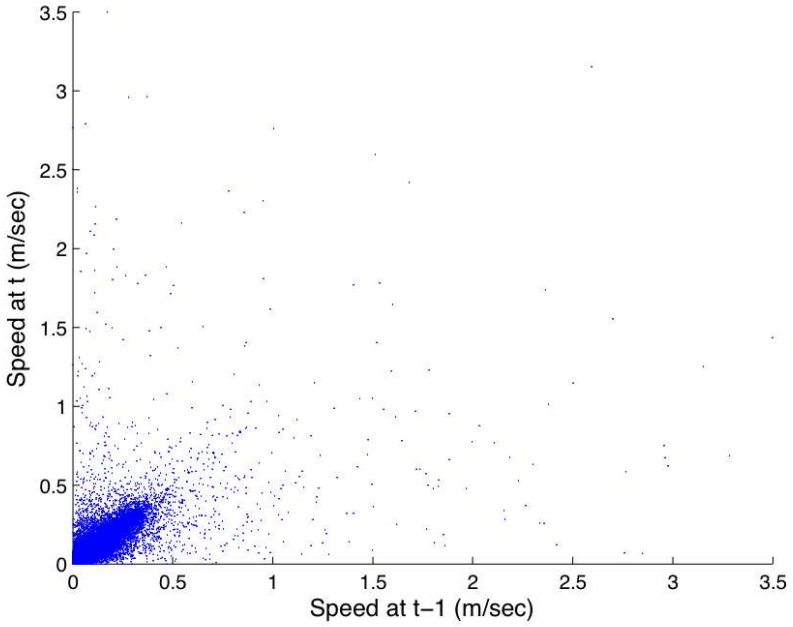


Figure 3: Correlation of current speed with the speed at the previous sample

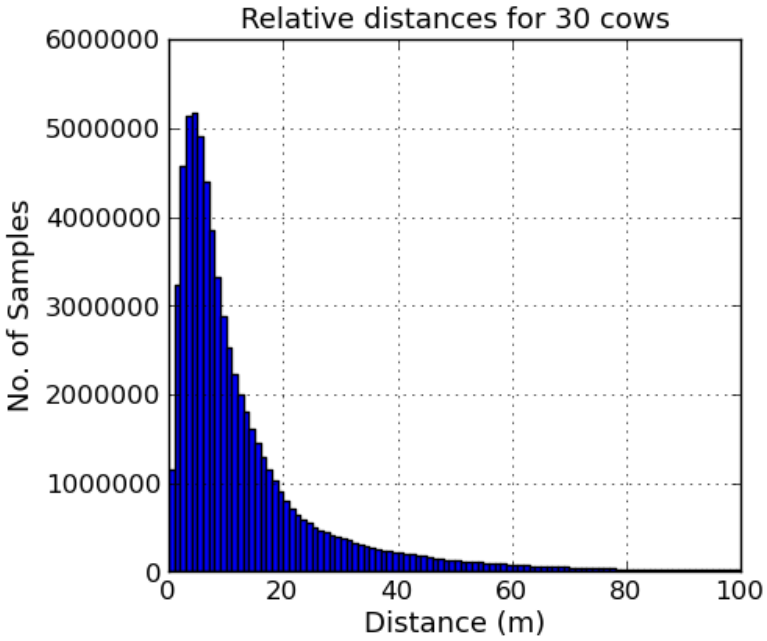


Figure 4: Relative distances between pairs of cows .

energy consumption at any time t during the deployment as

$$E_{gps} = [P_{gps}^{ON} \times DC_{gps} + P_{gps}^{OFF} \times (1 - DC_{gps})] \times t \tag{1}$$

where DC_{gps} is the GPS duty cycle, and P_{gps}^{ON} and P_{gps}^{OFF} are the power consumption of the GPS module in active and sleep mode respectively. We track the radio energy consumption as the sum of the transmission, reception, and listening energy values respectively [JRO08].

The MCU power consumption is heavily dependent on the GPS and beaconing strategies, as the MCU must power on when either the GPS module or radio are active. In general, the MCU duty cycle and energy consumed are expressed as:

$$\begin{aligned} DC_{MCU} &= DC_{gps} + DC_{radio} - overlap \\ E_{MCU} &= [P_{MCU}^{ON} \times DC_{MCU} + P_{MCU}^{OFF} \times (1 - DC_{MCU})] \times t \end{aligned} \tag{2}$$

where DC_{radio} is the duty cycle of the radio, and *overlap* denotes any overlap between the on-times for the GPS and radio. The only other component that keeps the MCU in active state is writing position data to the SD card, which occurs after a GPS lock operation. The SD write operation takes only a constant 5ms that is independent of the GPS or radio duty cycle. Because SD writes do not affect the duty cycling strategy and are negligible in duration compared to GPS locks, we disregard SD operations for our duty cycling analysis. The remaining node components also have a constant energy consumption. Sections 4 and 5 elaborate on MCU energy consumption further for specific duty cycling strategies.

3.3 GPS Duty Cycling Dynamics

GPS provides the most reliable absolute location estimate outdoors, with an average uncertainty[2] of +/- 5m. However, GPS's high power consumption demands that it is duty cycled in order to achieve the target lifetime. Whenever the GPS is active, the localization uncertainty is equal to the GPS error U_{gps}, which depends on the number of satellites currently visible, the satellite constellation, and the receiver clock error.

Whenever the GPS is powered off at time t_k, and until it next acquires lock, the radius of the position uncertainty region grows progressively according to

$$U(t) = \overline{s} \times [t - t_k] + U_{gps}(t_k)$$

where $\overline{s}$ is the assumed speed.

In a GPS duty cycling regime, the GPS module does not output a valid position estimate immediately after being turned on. It enters a state where the internal filter is initializing. This is referred to as *acquiring lock* and is indicated by a flag in the output data packet. Once it has lock, the GPS reports horizontal dilution of precision (HDOP), an indicator of error in the horizontal plane, and P_{acc}, which is the instantaneous error of the GPS position estimate. This value tends to start high, perhaps several tens of metres, and falls over consecutive samples to a few metres. The whole process of acquiring lock and an estimate with acceptable P_{acc} is simply referred to as obtaining a GPS *lock*.

Since we require $U < AAU$, and the GPS has a finite lock time, we need to start the lock process when $t = T_{off}$ where

$$T_{off} = \frac{AAU - U_{gps}(t_k)}{\overline{s}} - t_L \tag{3}$$

[2]The GPS chip uncertainty can vary between 2-15m depending on weather conditions, geographic region, and antenna size.

is the GPS off time. Once the GPS module acquires lock at the target position accuracy[3] $U_{gps} = P_{target}$, the GPS module is powered off again and the cycle repeats. It is clear in Equation 3 that t_L and U_{gps} are highly dependent on the GPS module behaviour. The remainder of this section characterises two GPS modules to expose their lock time and uncertainty features for use in our duty cycling algorithm. Section 4 focuses on how AAU and $\overline{s}$, the controllable parameters in our model, can be set for energy-efficient localisation.

3.4 GPS Model

Any GPS duty cycling for reducing energy consumption may affect the GPS lock time. The GPS lock time tends to increase when the GPS module is powered off for longer. To investigate this effect, we conducted two experiments for two generations of GPS modules: the u-blox generation 4 TIM4-H [uba] and the ublox generation 5 NEO5-G [ubb]). The NEO5-G provides a kickstart feature that is meant to significantly reduce the time to first fix by processing weak signals. Each experiment had a duration of one week in which the GPS was turned off for a random amount of time t_O ranging between 10 and 1800 seconds. Our analysis tracks two metrics: (1) the lock acquisition time t_{a}, which is the time that it takes the GPS module to report the first HDOP and P_{acc}; and (2) the position lock time t_{p}, which is the time after lock acquisition required for the P_{acc} to achieve a user-defined acceptable value P_{target}. The overall GPS lock time t_L is simply the sum of t_{a} and t_{p}.

The empirical data from these 1-week experiments for t_{a} is shown as a scatter plot in Figure 5. The results for the u-blox generation 4 module (Figure 5(a)) shows a linear cluster of points for the lock acquisition time for GPS off times less than 400 seconds, represented by

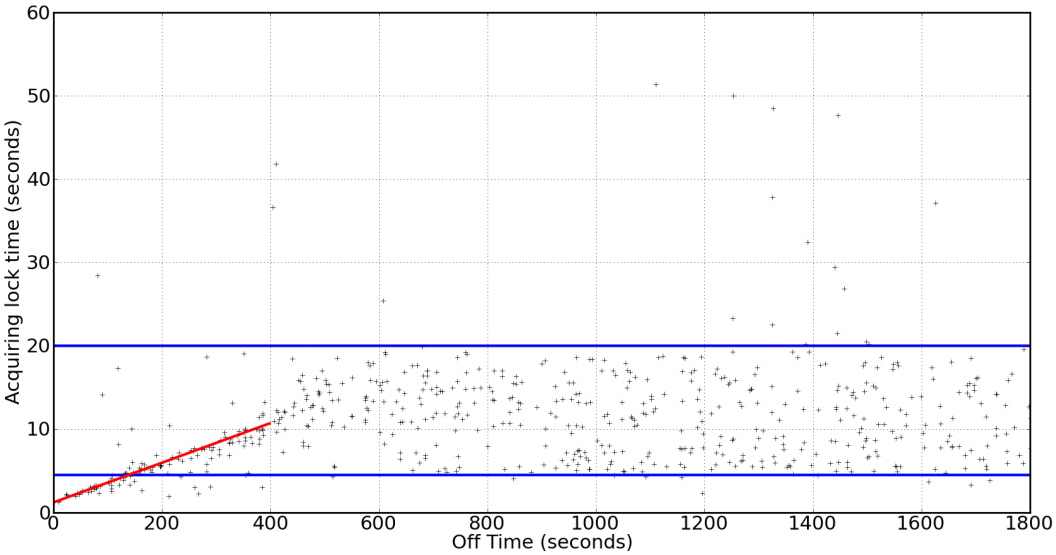
$$t_{\mathrm{a}} = 0.0238 t_O + 1.21$$

with a superimposed Poisson random process. For off times greater than 400 seconds, the lock acquisition times appear to be randomly distributed between 4.5 and 20 seconds. The performance of the generation 5 u-blox module (Figure 5(b)) exhibits much better lock acquisition time that is randomly distributed between 0 and 3 seconds, with a small number of outliers.

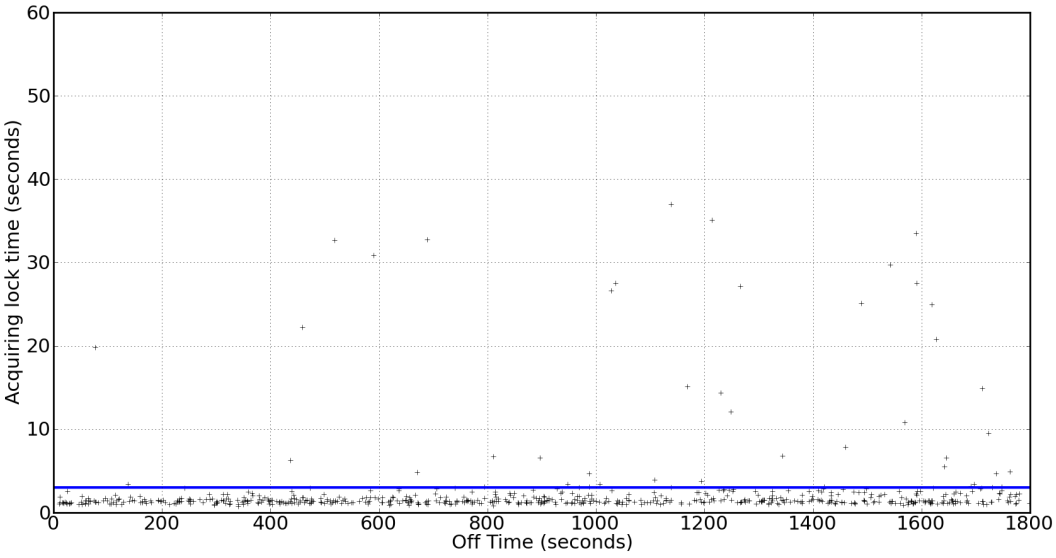
After the GPS module acquires its first 3 dimensional lock, it may take several seconds for P_{acc} to drop to the acceptable P_{target} value. In our experiments, we set P_{target} at 5 meters, which is the best accuracy that our GPS module consistently provides, and we observed the evolution of P_{acc} starting from the time of first lock acquisition and until it reaches P_{target}. The results are shown in Figure 6. There is a notable high variance in P_{acc} evolution for both modules, with a concentration of measurements at the lower end of the P_{acc} scale and a few large P_{acc} measurements contributing to a higher mean value. In the first few seconds after lock acquisition, both GPS modules exhibit a sizeable decrease in P_{acc}. This reduction flattens out thereafter, suggesting that there is a limited benefit in keeping the GPS module on for longer. The u-blox generation 4 (Figure 6(a)) P_{acc} at lock acquisition time is much lower than the generation 5 module, starting at an average of about 22m and falling to about 6m after a few seconds. The generation 5 module mean P_{acc} starts much higher, at about 75m, at first lock acquisition, and starts to match the generation 4 module after 5 seconds. This effect reduces the overall benefit of the generation 5 module in terms of overall lock time.

We now turn our attention to the GPS module uncertainty U_{gps}. This depends on the P_{target} setting. At any given time, a GPS lock is useful if the P_{acc} is less than both the node's current uncertainty and the AAU. The choice of an optimal P_{target} value involves a tradeoff: a lower P_{target} requires the GPS to stay on for a longer time but leads to a lower uncertainty; a higher P_{target} requires shorter on time for the GPS module but yields higher uncertainty. Quantitatively,

[3] P_{target} is a user-defined acceptable value for the GPS module error, which Section 3.4 explores in detail.

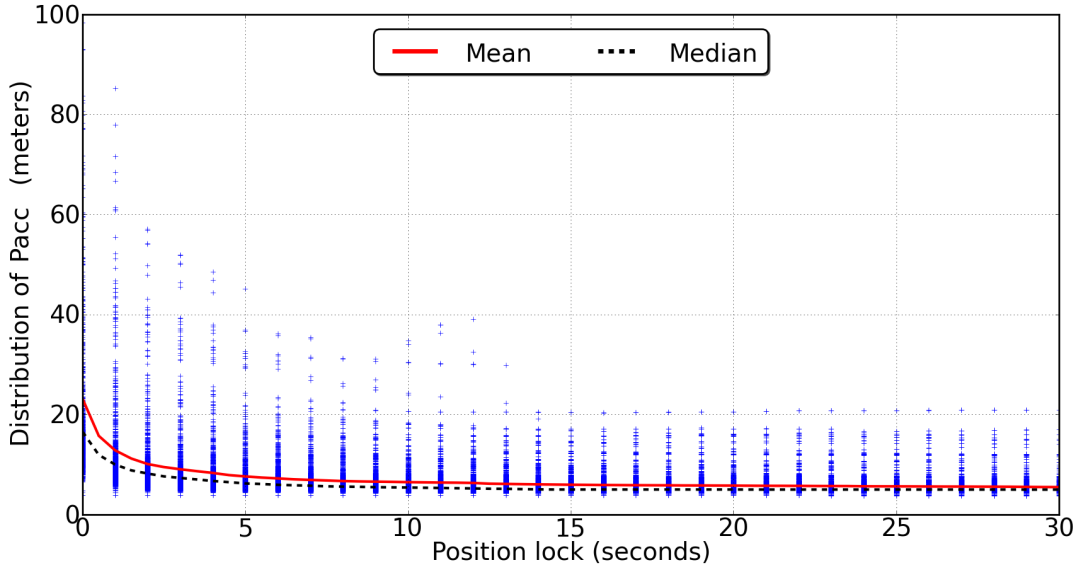


(a) Lock acquisition time for u-blox version 4

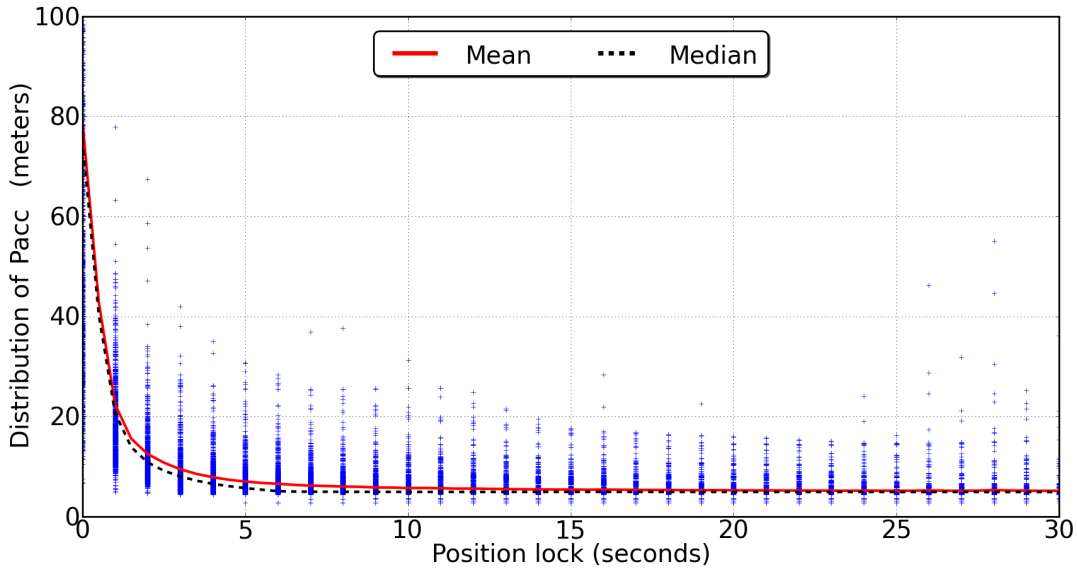


(b) Lock acquisition time for u-blox version 5

Figure 5: Lock acquisition time as a function of GPS off time for 2 versions of the u-blox GPS module.

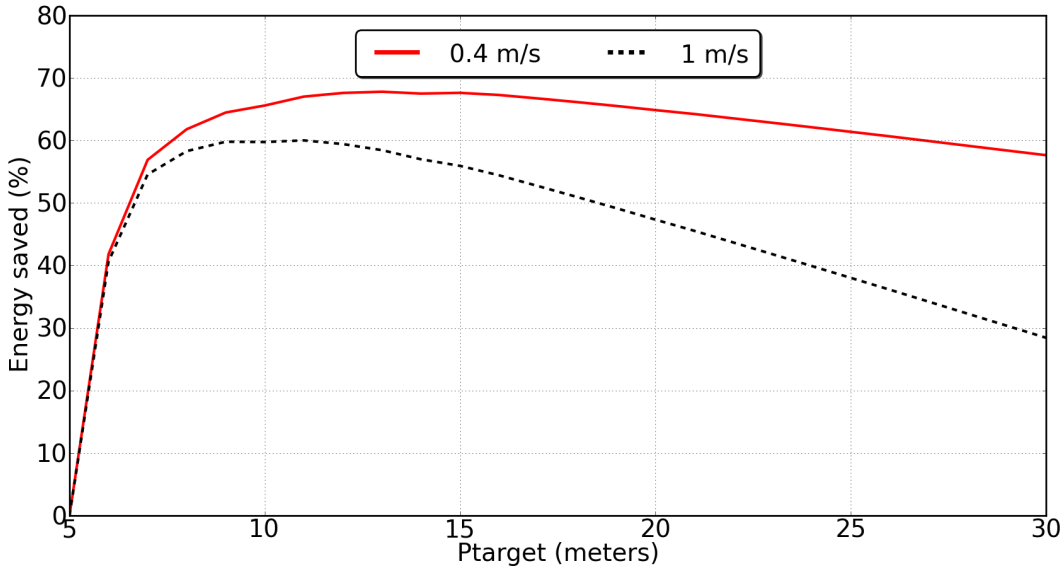


(a) Position lock time for u-blox version 4

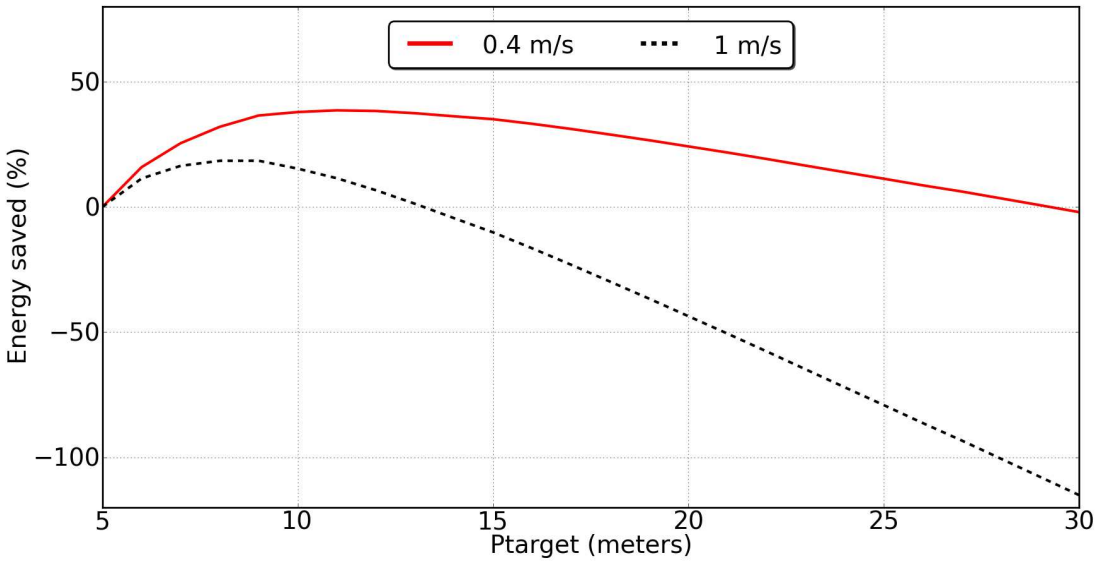


(b) Position lock tim for u-blox version 5

Figure 6: Position lock time as a function of GPS off time for 2 versions of the u-blox GPS module.



(a) Energy saved for u-blox version 4



(b) Energy saved for u-blox version 5

Figure 7: Energy savings as a function of HDOP for 2 generations of the u-blox GPS module. The energy savings are relative to a P_{target} setting of 5m. Curve peaks show the setting of P_{target} that maximises the off time for the GPS module. For the generation 5 module, setting P_{target} above 13m costs more energy, as it results in a net increase in the GPS on time.

setting P_{target} at a value M meters larger than the minimum achievable P_{target}^{min} reduces the GPS on time by X seconds, where $X = f(M)$ is determined by the results from Figure 6. Since U_{gps} is equal to P_{target} when we decide to power off the GPS, the resulting off time of the GPS becomes:

$$T_{off} = \frac{AAU - (M + P_{target}^{min})}{\overline{s}} - (t_L - f(M))$$

Reordering the expression yields

$$T_{off} = \frac{AAU - P_{target}^{min}}{\overline{s}} - t_L + f(M) - \frac{M}{\overline{s}}$$

where the first term represents the GPS off time using the lowest achievable P_{acc}, and is independent of M. The second term represents the additional amount of time that can be gained (or lost) per GPS lock by setting P_{target} higher than P_{target}^{min}. The problem of maximising the GPS off time then simply requires the selection of the value of M that maximizes

$$f(M) - \frac{M}{\overline{s}}$$

To investigate this tradeoff, we use the data log of the GPS module experiments in the previous section to determine the value of P_{target} that would minimise the energy consumption of our nodes. We curve fit the plot of the mean position lock times relative to P_{acc} in Figure 6. The resulting functions are:

$$f(M) = -0.927 + \frac{15.3}{M - 5.03}$$

for version 4 and

$$f(M) = 1.45 + \frac{7.40}{M - 4.85}$$

for version 5.

Figure 7 shows the results of this analysis. Since maximising the GPS off time is dependent on the assumed speed, we consider assumed speeds of 0.4m/sec and 1m/sec. The baseline energy consumption is for a P_{target} of 5m, and all energy savings are shown relative to this baseline. In Figure 7(a), setting P_{target} at 13m for an assumed speed of 0.4m/sec enables the GPS module to remain off for nearly 70% longer than for a P_{target} of 5m. Revisiting Figure 6(a) clarifies this effect. The position lock time for a P_{target} of 13m is only 2 seconds, while it may take the GPS module more than 30 seconds to achieve an accuracy of 5m. Although setting P_{target} at 13m means that the first term in T_{off} is $\frac{(13-5)}{0.4} = 20$ seconds less, the increase in the second term of T_{off} yields an overall increase in the off time for the GPS module. For an assumed speed of 1m/second, the value of P_{target} that yields the highest energy savings is reduced to 11m. Similar analysis for the generation 5 u-blox module shows a similar yet shifted trend, with optimal P_{target} values at 11m and 9m for assumed speeds of 0.4m/sec and 1m/sec respectively. The corresponding energy savings are smaller than for the generation 4 module, mainly due to the higher P_{acc} values in the first few seconds after lock acquisition. Interestingly, the savings diminish for longer P_{target}, and it becomes more energy expensive to increase P_{target} beyond some value than it is to keep P_{target} at 5m, particularly with the 1m/sec assumed speed. In other words, while powering off the GPS prior to reaching a P_{target} of 5m can save up to 70%, there are diminishing returns in powering off the GPS too quickly, since the reduction in position uncertainty fails to justify the energy cost of a GPS fix.

The analysis and experiments in this section have shown that the GPS modules we have evaluated follow a similar trend for maximising their off time. We define a common characteristic curve of the form

$$f(M) = a + \frac{b}{M - c}$$

Figure 8: A cow with our smart collar

where a, b and c are module-dependent constants. This curve can be used to maximise the off time of any GPS module. Finding the maximum value of $f(M)$ determines the P_{target} setting for maximising the off time of the GPS module, which correlates highly with energy savings for the node. We use these lessons in developing our duty cycling strategy in the following sections.

4 GPS duty cycling

The previous section has explored the dependency of position uncertainty on GPS module behaviour. This section formalises a GPS duty cycling strategy and explores its dependencies on the controllable factors in Equation 3, namely assumed speed and AAU. The section is structured as follows. Section 4.1 motivates cooperative localisation through a cattle tracking application that uses GPS-enabled collars. Section 4.2 formalises our duty cycling algorithm. Section 4.3 explores the effect of AAU, while Section 4.4 presents 3 different models for the assumed speed. Finally, Section 4.5 evaluates the performance of the algorithms and their dependency on uncertainty bound and speed models.

4.1 Motivating Application

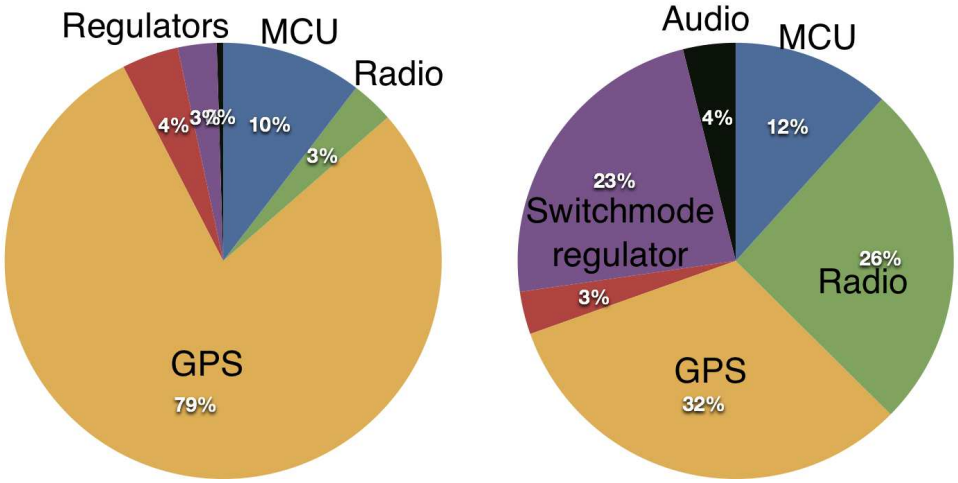
We consider an outdoor location monitoring application for tracking cattle using smart collars, as shown in Figure 8, that contain wireless sensor nodes and GPS modules. The goal is to track cow positions, and in some cases, enforce exclusion zones within the paddock, in effect a virtual fence [BCPR04, WCH+07]. Virtual fencing is useful for managing cattle in vast grazing lands, such as in Australia, where farms can reach the sizes of small countries, and it is not economically feasible to install physical fences in the whole area.

The error tolerance σ^* for this application is 5%. The target node lifetime is 3 months, which is the interval at which the animals are brought in for health checks, treatment and sorting. Achieving this lifetime is a challenge because the GPS module on each node has a large current draw [4]. Our cattle monitoring application and cow collar setup are fully described

[4]Smart phones that include GPS modules involve similar considerations, albeit with more regular opportunities

Component	Active ModePower (mW)	Sleep Mode Power (mW)
MCU	18	1.2
Radio (TX)	79.2	0.00825
Radio (RX)	29.7	0.00825
GPS	165	0.215
Switch-mode Regulator	6	6
Linear Regulator	3.3	3.3
Audio	3	0

Table 1: Active and sleep mode power consumption for collar components



(a) Always on GPS

(b) GPS with a 5% duty cycle

Figure 9: Impact of GPS duty cycling: Reducing GPS duty cycle from 100% to 5% reduces the node power consumption by 7.5 times. The radio has a 10% duty cycle in performing low power listening for both cases.

in [WCH+07]. We limit the discussion here to a high level description of the collar setup and the energy consumption of each of its components.

The collars incorporate a Fleck node, a GPS module, and an audio board for generating audio cues to indicate to animals that they are crossing a virtual fenceline. The Fleck node itself comprises an Atmel-1281 MCU and Atmel RF212 radio. Four D-Cell batteries in series provide power to all the collar components via several switch-mode regulators. The weight of these batteries yields a collar weight that approaches the upper limit of 1.5 Kg set by an animal ethics committee in Australia, so larger batteries are not an option.

The active and sleep mode power consumption of these components are shown in Table 1. The power consumption of the GPS board is the largest by far. To put things in perspective, Figure 9(a) illustrates a scenario with the GPS constantly active. Using our collars with a 10% radio duty cycle, based on a 6ms check interval and 60ms sleep interval, and an always-on GPS board, the GPS accounts for 88% of the power consumption of 209 mW and limits the nodes' lifetime to 19 days. Figure 9(b) sets the GPS duty cycle at 5%, while maintaining the same setup for all of the collar components. In this scenario, the GPS accounts for about a third of the overall node power consumption of 25mW, which extends the nodes' lifetime by a factor of 7.5 (nearly 6 months). This confirms the importance of duty cycling the GPS on the mobile nodes. In both scenarios, the MCU remains on only as required to service to the GPS or radio, which couples its duty cycle to these components' duty cycles.

for recharging batteries.

```
loop {t}
  choose_speed()
  if GPS == LOCK then
    position ← GPS position
    U ←U_gps
    lastlocktime = now()
    GPS = OFF
  else
    U ← U + (t * S_c)
    if GPS == OFF and U+(t_L*s̄)> AAU then
      GPS = ON
    end if
  end if
end loop
```

Figure 10: GPS duty cycling strategy

4.2 Duty Cycling Algorithm

Figure 10 illustrates the algorithm for duty cycling the GPS module. The algorithm uses the choose_speed() function to select the speed model. At each time step the algorithm checks if the GPS has lock and if so it obtains the position, sets the uncertainty to U_{gps}, then powers off the GPS. Otherwise, it grows the uncertainty according to the current speed S_c. Once the uncertainty approaches AAU, it turns on the GPS and the process repeats.

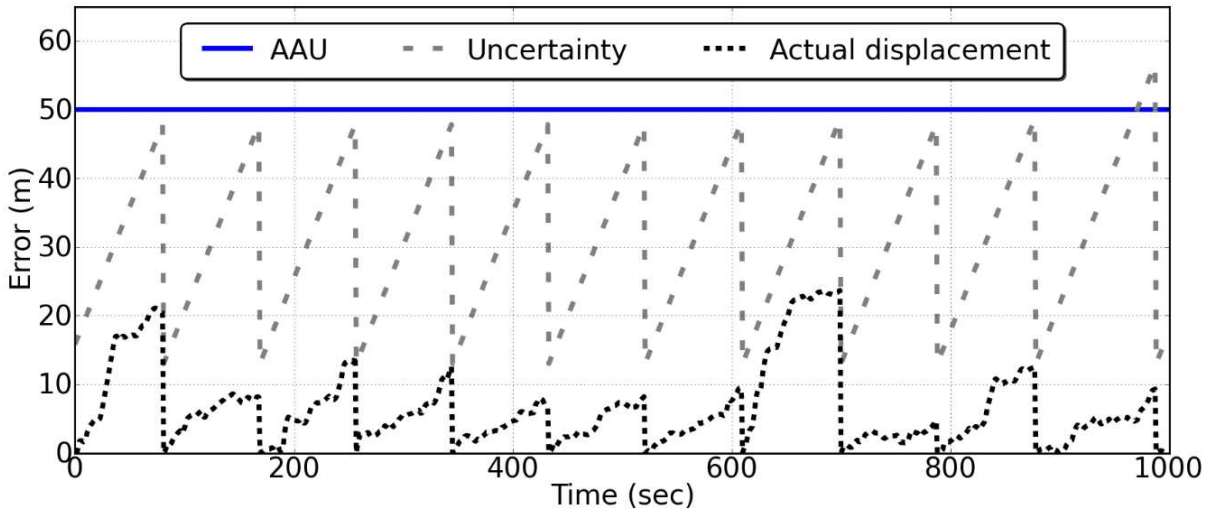
The value of $\overline{s}$ can be specific to each mobile node, and can be a constant, or vary as a function of time and space. The lock time, t_L, has a weak dependence on the length of time that the GPS has been off, which we can control. It also depends on factors beyond our control, such as the time varying satellite configuration and the deployment environment. In practice, a "best estimate" of t_L is used or updated in real-time on the node based on the measured time taken to acquire lock. Nevertheless, long lock times will occur with low probability, which means that the estimated uncertainty will occasionally exceed AAU.

4.3 Uncertainty Bounds

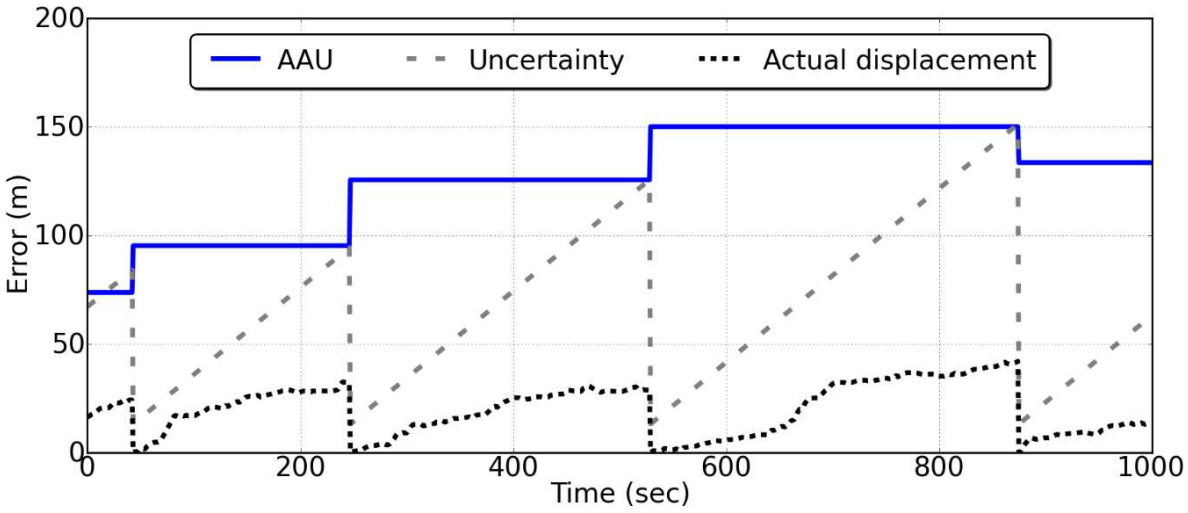
Most positioning applications require a fixed uncertainty bound. For a specific class of applications, such as virtual fencing, the AAU can vary dynamically during the deployment according to application state. For instance, in the virtual fence application, the AAU can vary as a function of the node's position and is the distance of the node from the virtual fence line. If the node is 1000 meters away from the fence, we simply require that the next time the GPS makes a measurement, it will not have crossed the fence line. If we assume the node has a maximum speed of 1m/s (which is true 95% of the time), then a node can keep its GPS off for more than 985 seconds, with a conservative estimate of 15 seconds for GPS lock time. A node that was 100m away from the fence at the last lock can keep its GPS powered off for 85 seconds. Comparatively, a fixed AAU of 50m would result in a GPS off time of only 35 seconds. The sizeable increase in GPS off times relative to a fixed AAU can greatly reduce the average power consumption for applications with dynamic AAU.

Figure 11(a) illustrates the effect of AAU on the growth of the estimated uncertainty, with P_{target} set at 13m, the optimal value for the u-blox generation 4 module. The actual displacement in this plot indicates the ground truth displacement [5] of the mobile node since the last GPS lock.

[5]Based on 1 Hz GPS data.



(a) Uncertainty behaviour for a static AAU



(b) Uncertainty behaviour for a dynamic AAU

Figure 11: Effect of AAU on uncertainty behaviour

In the real system actual displacement is not available — position errors are only discovered when the node acquires its next GPS lock. The purpose in tracking the actual displacement here is to determine how well our uncertainty estimates the nodes' mobility.

We choose an assumed speed $\overline{s}$ of 0.4 m/sec for the nodes, which is the average node speed in the cow dataset, and the estimated uncertainty grows linearly while the GPS module is powered off. When uncertainty approaches AAU, the node activates the GPS and keeps it on until P_{acc} reaches P_{target}, resulting in periodic duty cycling of the GPS. A total of 12 GPS locks occurred in this time window.

In Figure 11(b) the AAU is varied according to the distance between the last observed position of the mobile node and the virtual fence line. While the user sets an initial AAU of 50m, the AAU rises to about 80m at the first GPS lock, reflecting how far the cow is from the virtual fence. The AAU increases to further to 97m and 125m at the next two GPS locks. The higher AAU allows the node to keep its GPS module powered off for longer, reducing the number of GPS lock attempts to 4, with the actual displacement never exceeding the uncertainty.

Static	*Dynamic*	*Probabilistic*
$S_c = \overline{s}$	if (s(t)> $\overline{s}$)	i=t-lastlocktime
	S_c=s(t)	if (i==0)
	else	S_c=s(t)
	$S_c = \overline{s}$	if ($S_c > \overline{s}$)
		$P = t_{22}$
		else
		$P = t_{11}$
		else
		$S_c = P \times S_c + \overline{s}(1 - P)$
(a)	(b)	(c)

Figure 12: The choose_speed() function for each speed model

Assumed Speed (m/s)	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0	2.2
t_{11}	0.82	0.89	0.92	0.93	0.94	0.95	0.96	0.96	0.97	0.97	0.97
t_{22}	0.86	0.85	0.84	0.84	0.83	0.82	0.82	0.81	0.81	0.81	0.81

Table 2: Probabilities of remaining in the same state (moving or stationary) as the assumed speed varies.

4.4 Speed models

The assumed speed is a critical parameter. If it is too low, then the AAU bound will be violated when the animal moves quickly. If it is too high, the GPS off times will be very short which is costly in terms of power. Choosing the right speed requires a speed model, which is highly dependent on the application. The data in Figure 2 indicates an average cow speed of only 0.4 m/sec, and 95% of the time the cows move at 1m/sec or less, with an absolute observed maximum of 3.5m/sec.

We explore three models for the assumed speed $\overline{s}$, which are summarised in Figure 12:

1. Static: based on a constant assumed speed
2. Dynamic: based on setting the assumed speed as the last observed speed of the mobile node, read from the GPS speed value from the GPS module at last lock
3. Probabilistic: based on setting the assumed speed on the basis of the last observed speed and a state model of the mobile node

The static and dynamic speed models are application-independent. The static speed model assumes a low variance from an average speed and uses this estimate for the entire deployment. The dynamic speed model assumes a high correlation between the most recent speed measurement and the current speed estimate, i.e. persistence. The probabilistic speed model also relies on persistence, but assigns application-specific probabilities for decaying speed estimates. For instance, previous work has modeled animal mobility as a 3-state Markov process [GPC+09]. Unlike some other mobility models in the literature for mobile phone or car applications [LNR04], the animals do not follow specific routes, rather they spend a lot of time in random foraging behaviour.

The probabilistic model in this paper is a simplification of the Markov model of cow speed in [GPC+09] where the animal has a slow- and a fast-moving state, states 1 and 2 respectively. State transitions from state i to state j are referred to as t_{ij}. Using data from the 2-day cattle

position trace, we determine the transition probabilities for each state based on the observed speed. Table 2 shows the probabilities for remaining in either the stationary or moving state based on the assumed speeds. There is clearly a high degree of persistence for both states. Nodes are more likely to stay in the stationary state (state 1) for higher assumed speeds, and they are slightly less likely to stay in the moving state (state 2) with higher assumed speeds. These transition probabilities are then stored at the mobile nodes.

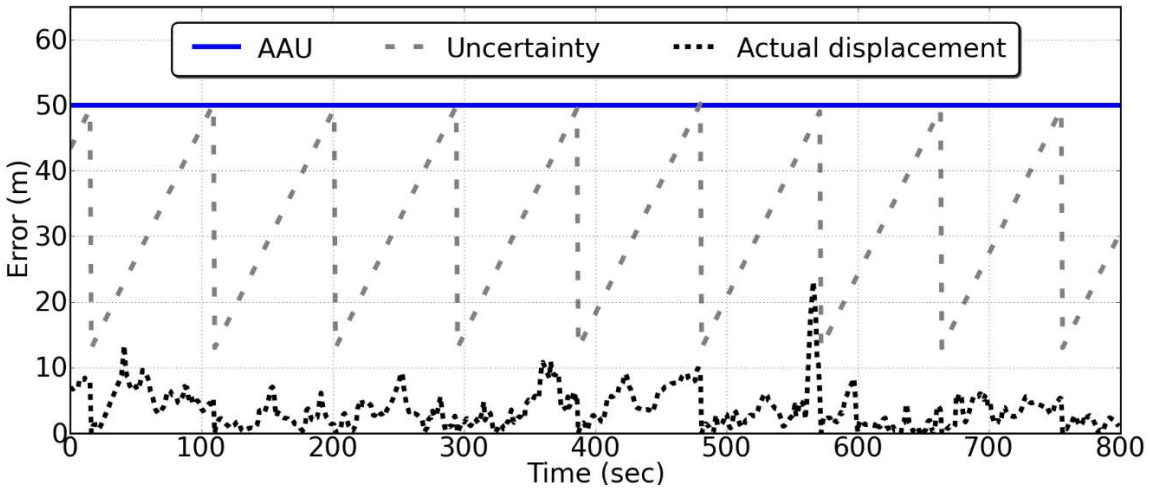
Whenever the GPS acquires lock in the probabilistic model, we set the assumed speed to the last observed speed, as for the dynamic model. When the GPS is powered off, the assumed speed gradually moves toward $\overline{s}$ using a first-order filter. The time constant is a function of the transition probability of the initial state. The reasoning is that as the time since the last speed observation grows, its significance decreases until at some point, and without any other data we revert to a constant assumed speed.

Figure 13 illustrates the effect of each speed model on uncertainty estimation for a constant AAU. All three models use the same $\overline{s}$ of 0.4 m/sec. The uncertainty in the constant speed model (Figure 13(a)) grows regularly according to $\overline{s}$, resulting in 9 GPS locks. For the dynamic speed model in Figure 13(b), the uncertainty growth rate follows the changes in $\overline{s}$. For instance, at 100 seconds, the node acquires lock and observes that the current speed is higher than $\overline{s}$, and as a result, it increases $\overline{s}$ to better model the actual motion. At the following GPS lock, the same node observes a further increase in the measured speed, and increases $\overline{s}$ again for its speed estimate. The dynamic model adapts better to node speed changes while increasing the number of necessary GPS locks to 12. Finally, the probabilistic model results in 9 lock attempts, yielding similar performance to the static speed model. This similarity is due to the GPS lock times in the probabilistic model in this example, which appear to align with speeds that are below the constant assumed speed, leading to the constant assumed speed being used for most of the time.

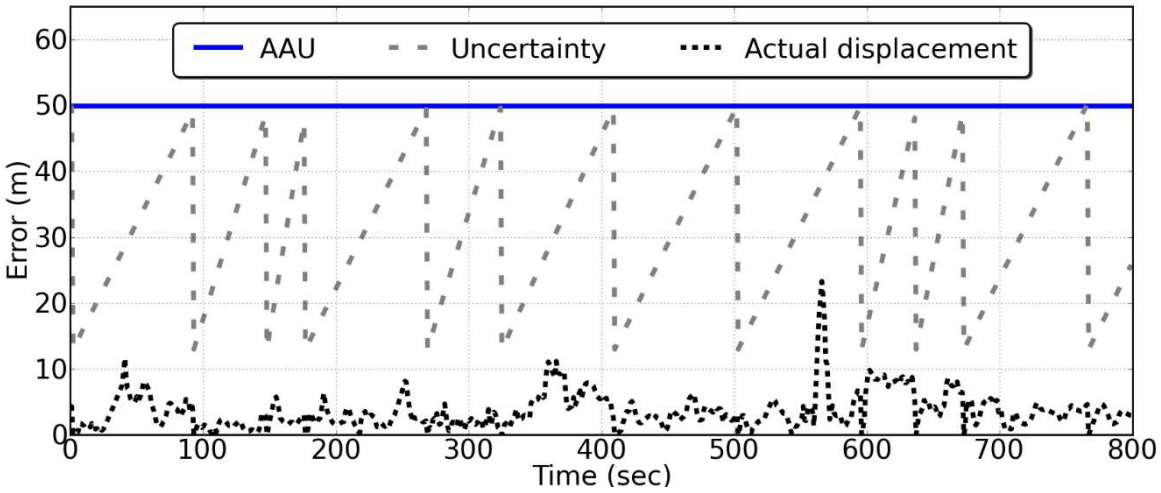
4.5 Performance Evaluation

To evaluate the impact of our GPS duty cycling strategy on position accuracy and energy efficiency, we have implemented the duty cycling strategy of Figure 10 in a detailed Python-based simulator that uses the cow position data described in Section 3.1. Each GPS module is represented by a class that implements a simple state machine which includes the probabilistic lock time model of Figure 5 and the preceding off time. When the GPS is turned off, the simulator evolves the uncertainty estimate according to the speed model. The simulator keeps the GPS module off for T_{max} seconds then turns it on to start the locking process. When the simulated GPS obtains lock, the true position is used to determine if the AAU constraint has been violated and to update the error rate σ. The simulator also updates the average power consumption for the GPS, radio, MCU and regulators based on the energy model in Section 3.2, which enables the computation of the node's overall power consumption.

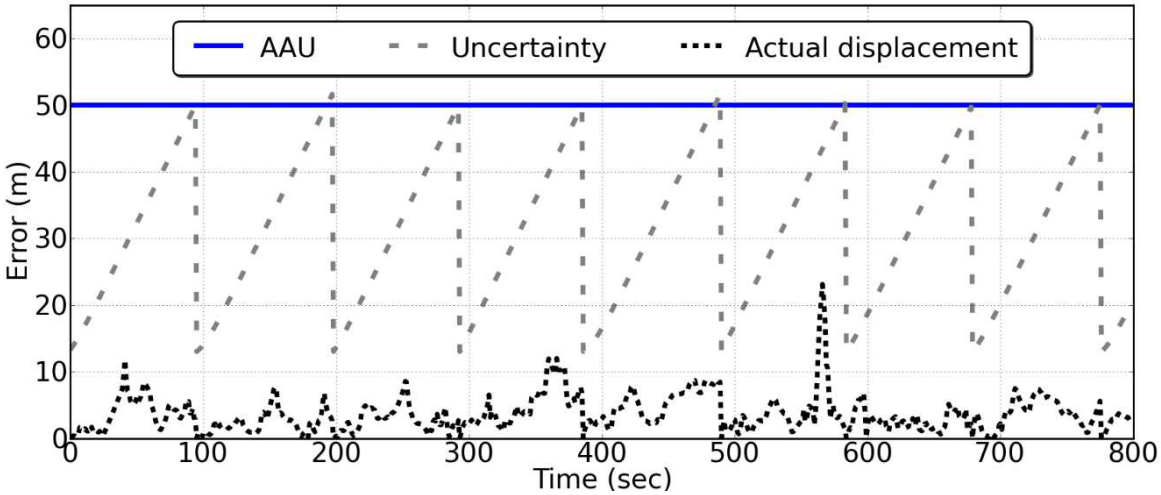
We initially fix the AAU to 50m and explore the impact of changing the assumed speed for each of the three speed models. The results are shown in Figure 14. As expected the GPS power consumption dominates the node power consumption in all cases. The GPS power consumption increases with assumed speed, since the faster growth in estimated uncertainty triggers more frequent GPS fixes. Because the GPS and radio timers fire completely independently, the *overlap* parameter for computing the MCU duty cycle is zero. The MCU power consumption increases with increasing GPS activity, as the MCU has to remain in active state for interacting with the GPS module. The radio power consumption, based on a low power listening duty cycle of 5% with 6ms check interval and 128ms sleep interval, is a small contributor to the overall node power consumption, since it only sends short data packets every minute back to the base station.



(a) Uncertainty behaviour for the static speed model



(b) Uncertainty behaviour for the dynamic speed model



(c) Uncertainty behaviour for the probabilistic speed model

Figure 13: Effect of speed model on uncertainty behaviour

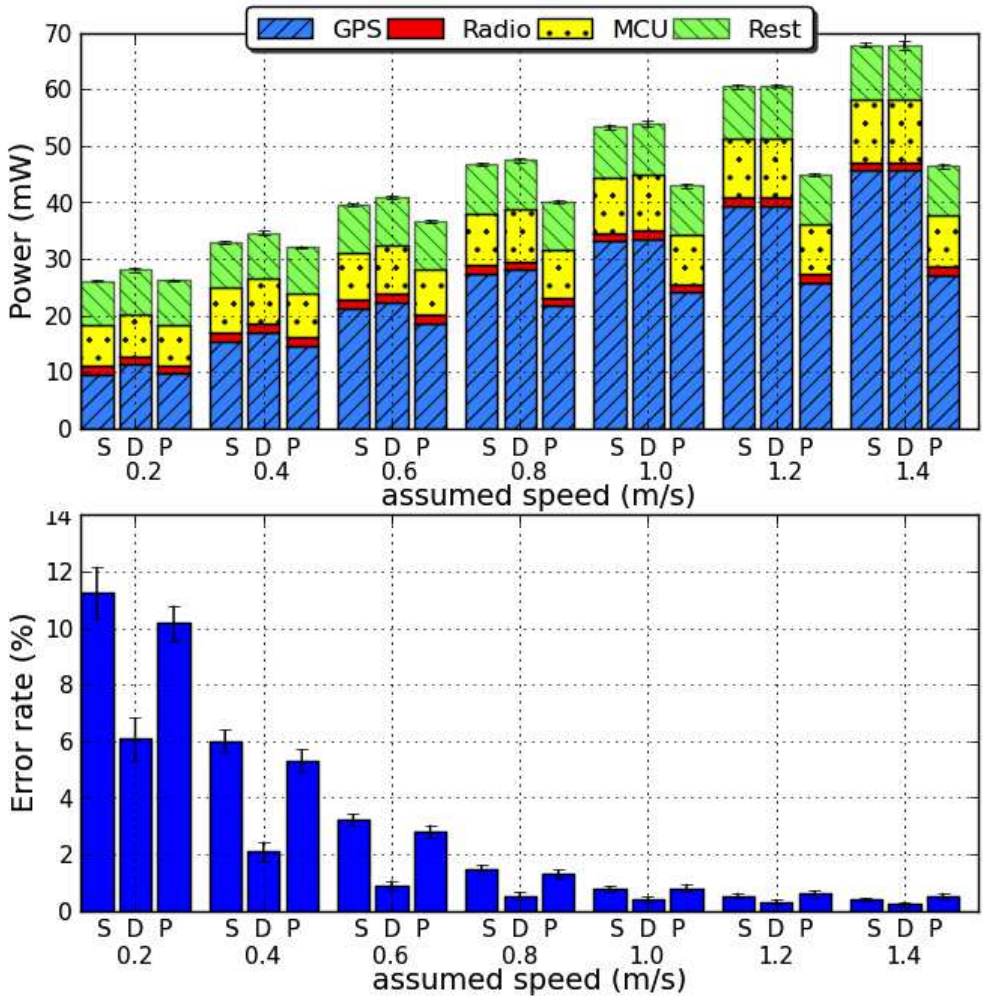


Figure 14: Power consumption and error rate for the 3 speed models (fixed AAU). S: static, D: dynamic, and P: probabilistic.

The GPS power consumption of the probabilistic speed model has the lowest dependence on the assumed speed, as it always uses the most recent observed speed estimate for most of the time. For low assumed speeds, the dynamic speed model has a slightly higher power consumption than the static speed model, because it occasionally uses a much higher assumed speed when it happens that the speed observed over the previous interval is high. As the assumed speed increases, both the static and dynamic speed models exhibit similar behaviour, as there are fewer instances where the dynamic model switches to current speed values. The dynamic speed model performs best in terms of accuracy, with its error rate σ decreasing steadily with increasing assumed speed, from about 6% at an assumed speed of 0.2m/sec to less than 0.3% for assumed speeds above 1.2 m/s. The probabilistic and static speed models have similar power consumption at low speeds, with a slight advantage in error rate for the static model. At higher speeds, the static model achieves lower error rate at the cost of higher power consumption, since it always maintains a conservative assumed speed, while the probabilistic model can often use a lower assumed speed according to its state machine.

Figure 15 shows the effect of enabling dynamic AAU in a virtual fencing application according to distance from the fence. The variance of results here, indicated by error bars, is noticeably higher than for static AAU, because of the disparity among nodes' distances to the fence during the deployment. The dynamic AAU reduces the GPS power consumption by more than half for

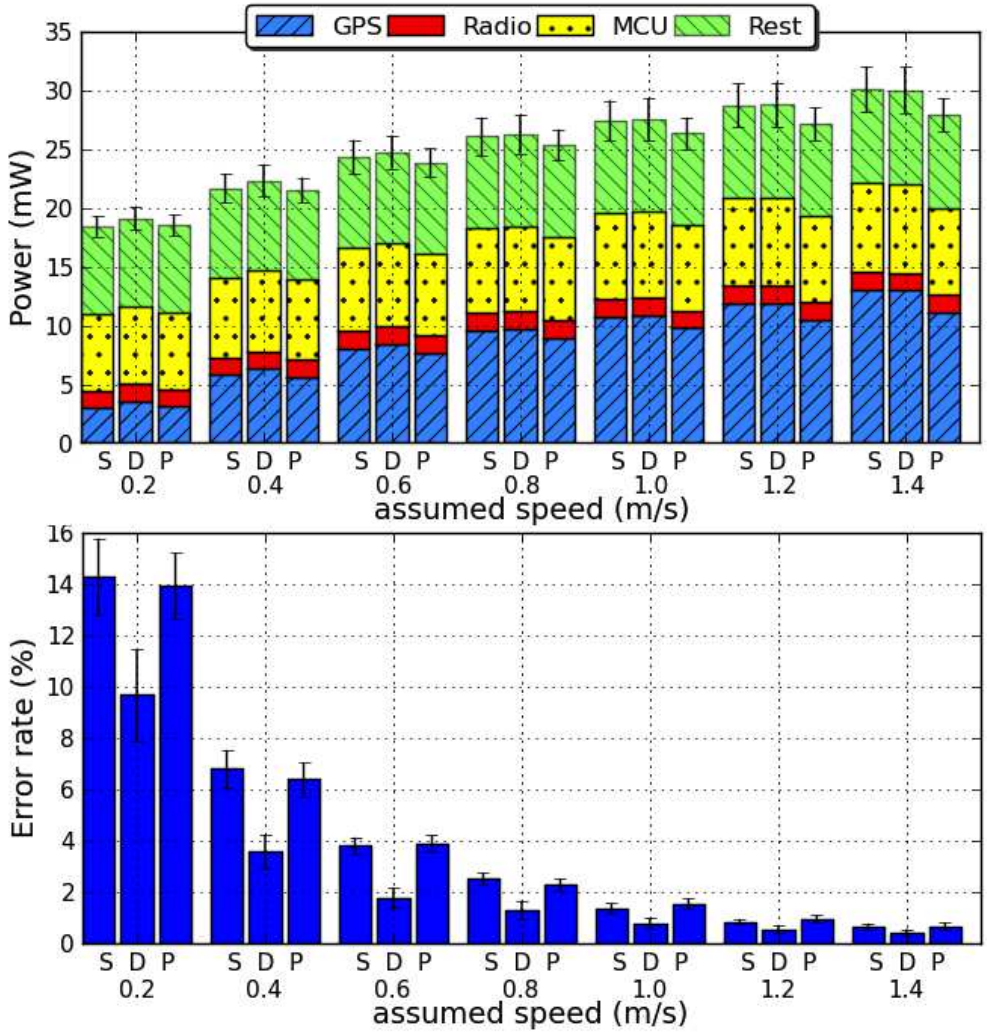


Figure 15: Power consumption and error rate for the 3 speed models (dynamic AAU). S: static, D: dynamic, and P: probabilistic.

low assumed speeds, and more than 70% for higher assumed speeds. For assumed speeds of 0.4m/s or less, the GPS power consumption accounts for less than the MCU or the other node components, at a cost of increased error rate. The overall node power consumption increases more linearly with assumed speed for dynamic AAU. At the highest assumed speed of 1.4 m/s, the overall node power consumption is about 40% of the static AAU case. The reduced power consumption of dynamic AAU comes at the cost of a higher error rate. This stems from the longer off times for the GPS modules, which increases the expected lock times and the instances where the GPS takes a very long time to obtain a fix. While dynamic AAU can yield significant energy savings at the cost of slightly higher error rates, dynamically changing AAU according to the distance from the fence is a specific feature of the virtual fencing application and similar tracking applications with known boundaries. The remainder of the paper focuses on static AAU, which is more representative of a wider class of GPS-enabled mobile applications.

5 Coupling GPS and contact logging

The previous section has presented a GPS duty cycling strategy which bounds position uncertainty to improve energy efficiency. Mobile nodes can also use non-GPS relative localization signals to reduce position uncertainty while the GPS module is powered off. For instance, neighboring nodes can share their location estimation over short-range radio communications. This strategy is particularly attractive for application classes where several mobile nodes cluster together, such as cattle tracking.

5.1 Overview

Consider that node A has powered off its GPS and is growing its estimated uncertainty as a function of time. When node A is some distance $\hat{r}$ from node B, node A receives node B's beacon. Node A can infer that node B is within a contact radius of R, where R is the conservative bound of communication range ($R \geq \hat{r}$ for all beacons).

The beacon from B also includes its last measured position x_k^B and its current uncertainty estimate U^B. If $U^B + R < U^A$ then node B is nearby and has a lower uncertainty than node A. In this case node A lowers its uncertainty estimate

$$U^A := U^B + R \tag{4}$$

Lowering the estimated uncertainty enables node A to keep its GPS off for longer, which in turn reduces its energy consumption. If all nodes run this algorithm, the energy-expensive GPS position lock is shared across the nodes. The fairness of the algorithm stems from the fact that when node A relies on node B for its position estimate, $U^A > U^B$. If the two nodes use the same assumed speed to grow their uncertainty, then node A will decide to turn on its GPS before node B in the next cycle, which allows B to rely on A for its position estimate in the next cycle[6].

If a node's estimated uncertainty approaches AAU, it turns the GPS on. If the GPS is acquiring lock when U^A is updated and if $U^A < AAU - \overline{s} \times \overline{t}_{lock}$ then node A should turn off its GPS, even without acquiring lock.

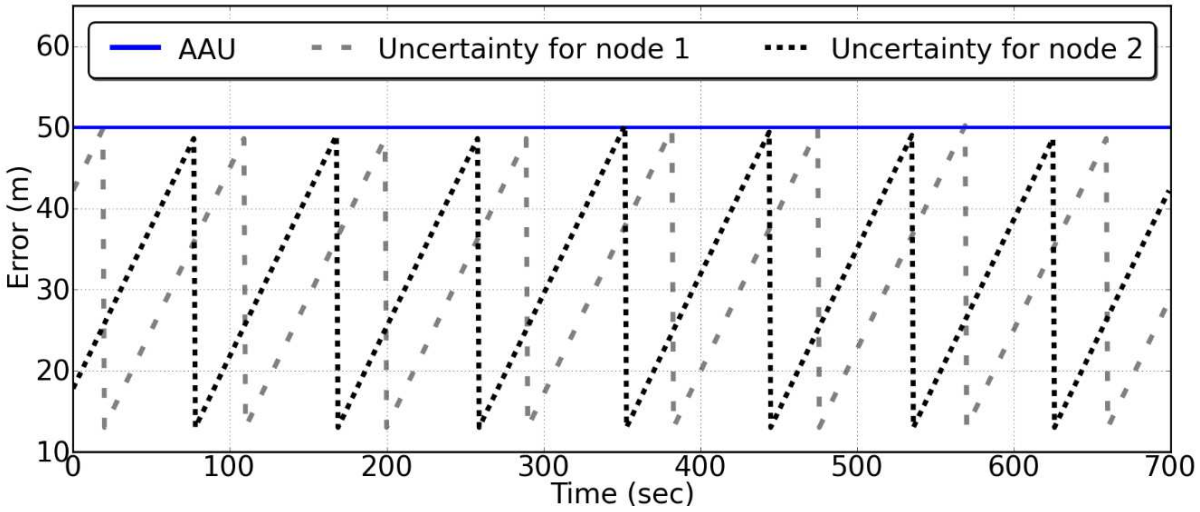
Figure 16 illustrates the impact of contact logging on GPS duty cycling for two nodes that are sending contact beacons every second. Without contact logging (Figure 16(a)), each node independently tracks its uncertainty estimate and acquires a GPS position fix whenever its uncertainty approaches AAU, resulting in 7 fixes for node 1 and 6 fixes for node 2. Using contact logging with R=10m, the nodes can reduce their GPS fixes to 5 and 4 locks in the same time window. Consider the plot at 480 seconds. The uncertainty for node 2 approaches AAU, so it powers on its GPS and obtains a position fix. Node 1, which is in the vicinity of node 2, can rely on the latter's recent GPS fix to reduce its uncertainty to the 10m contact radius plus U_{gps}. This example illustrates the fairness feature in our contact logging strategy, since nodes that recently acquire lock will have a smaller uncertainty estimate than neighbours, forcing another node to turn on its GPS in the next round.

In general, each node in a cluster of N nodes that are using our contact logging strategy would have a theoretical maximum off time according to:

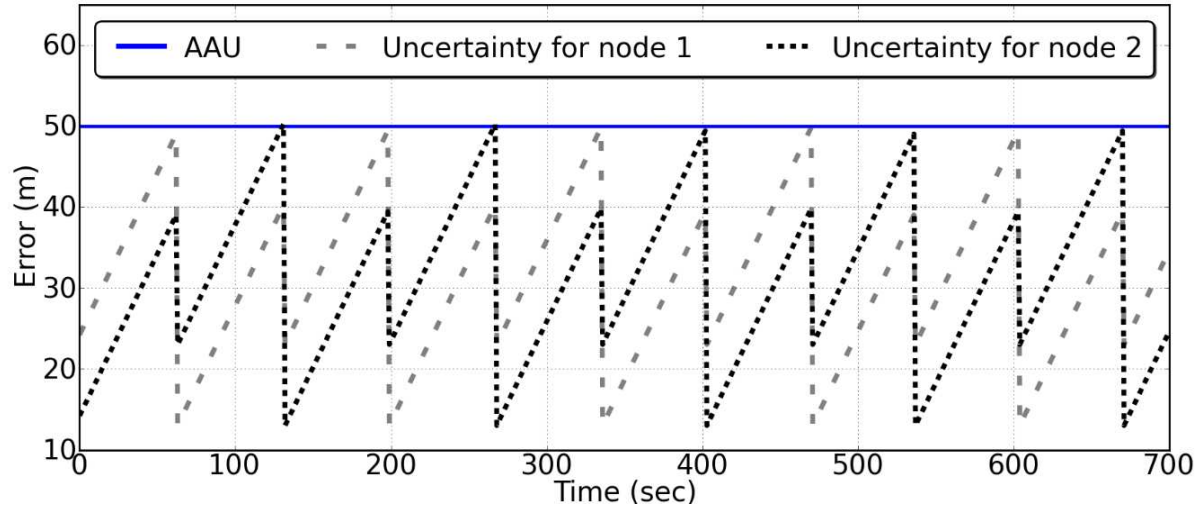
$$T_{off}^N = \frac{N(AAU - U_{gps} - R)}{\overline{s}}$$

However, achieving this theoretical maximum requires that the node GPS lock schedules are sufficiently desynchronised. For $N > 2$, contact logging may generate implicit synchronisation among several nodes that rely on a neighbour acquiring GPS lock. This can reduce the effective

[6]Using a variable speed model can result in exceptions to this trend, but contact logging will result in long-term sharing of GPS load if the assumed speed follows the same random distribution on all nodes.



(a) Error behaviour for GPS duty cycling



(b) Error behaviour for GPS duty cycling coupled with contact logging

Figure 16: Impact of contact logging on GPS duty cycles

GPS off time of a cluster larger than 2 nodes. The dynamic speed model does implicitly contribute to desynchronisation within a cluster, but its benefits are limited as the node movement within a group is typically similar to other nodes in the cluster. Section 5.4 proposes a simple backoff mechanism for avoiding beacon transmission synchronisation, but uncertainty cycle desynchronisation is an open issue that is beyond the scope of this work.

Our contact logging strategy depends on the settings of two variables: (1) the contact radius; (2) the contact beacon period. The contact radius specifies the value of R, which must balance the need to cover larger areas with a higher R and to reduce uncertainty with a lower R. The beacon period is the time between beacons at each node. It involves a tradeoff: short contact periods increase the likelihood of useful contact but incur a higher energy cost.

Section 5.2 presents simulations based on empirical data to determine the operating points for contact radius that balance these tradeoffs. Section 5.3 then investigates the impact of beacon period on the energy/accuracy tradeoff.

5.2 Contact Radius

5.2.1 Contact Radius Selection

We now explore the effect of the contact radius setting on performance. The main mechanism for setting the contact radius is to change radio receiver sensitivity and/or to change radio transmit power. Given that these settings will not provide a fixed or uniform signal transmission, the value of R conservatively reflects the maximum contact radius.

To accurately model the tradeoffs of contact logging, we extended our Python simulator with a Message class. Each instance of the Message class represents a contact beacon which enables communication between instances of the Collar class. A transmitted contact beacon is registered as received if the distance between the sender and receiver is less than the contact radius. The time slot for sending contact beacons and the order in which nodes send and receive contact beacons are randomized to avoid simulation seeding artifacts. Each beacon contains the following information: node ID, node position, and node uncertainty.

Figure 17 summarizes the impact of contact radius on node energy consumption and error rate for the entire cattle data trace. The contact beacon period is set to 1 second. The *overlap* parameter here for the DC_{MCU} is equal to DC_{gps}, as the frequent radio beacon transmissions and channel checking imply that the radio nearly always on when the GPS module is on. For all contact radii we considered, the error rate remains within the 5% tolerance for our application. In terms of energy efficiency, the optimal contact radius is 5m, which confirms the analysis in [JCDS10]. Although contact logging reduces GPS duty cycle by up to 60%, the overall node power consumption is only marginally reduced, as contact logging increases the radio activity and associated MCU activity. A contact radius of 20m provides the lowest error rate, as it effectively limits useful contact beacons to 1-hop. This is a direct consequence of adopting a P_{target} of 13m and an AAU of 50m, meaning any node logging useful contact would set its uncertainty to 33m. This node will not serve as a position reference for any other neighbour, because it can only provide these neighbours with a position estimate with uncertainty of 53m, which is higher than the AAU. For contact radii of 5m and 10m, multi-hop contact logging is possible, but it appears to have a slightly negative effect on error rate. As the contact radius is increased above 20m, the improvements in error rates move towards the plain GPS duty cycling case, simply because the nodes high uncertainty limits the portion of time they can rely on contact logging position estimates. The radio power consumption notably increases for larger contact radii, which increases the range of node transmissions and causes more overhearing. The conclusion from these results is that simple contact logging with a fixed radius and a 1 second beacon period does not yield worthwhile benefits over simple GPS duty cycling.

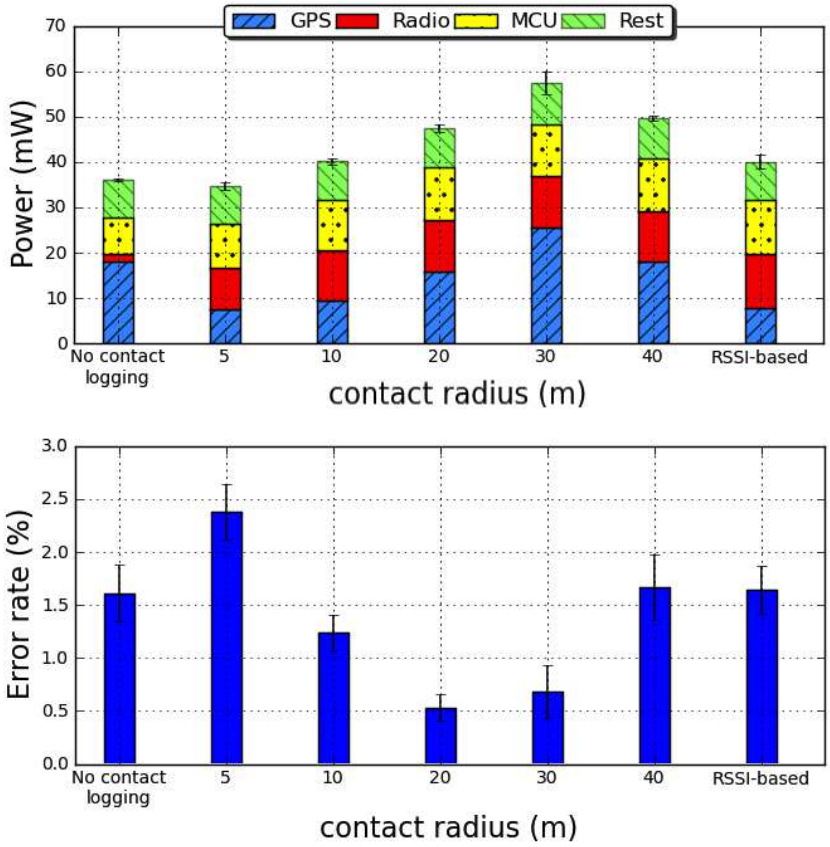


Figure 17: Effect of contact radius (static AAU)

5.2.2 RSSI-based Ranging

As mentioned above, estimating the distance $\hat{r}$ between two nodes as the radio range R is conservative. The use of received signal strength indicator (RSSI) can relax this estimate. While RSSI is a poor estimator of absolute range in mobile or non-line-of-sight environments due to multi-path signal propagation and interference, it still highly correlates with range between two nodes [SL06]. Here, we use RSSI to bound the range between two nodes as follows. Line-of-sight RSSI measurements with varying distance are collected prior to deployment in an open field. Let the line-of-sight RSSI value of two nodes that are $\hat{r}$ meters apart be X. If during a deployment, a node measures an RSSI of X from a neighbour, then the neighbour is at most $\hat{r}$ meters away.

Figure 18 shows 2400 line-of-sight RSSI measurements we collected for the Atmel RF212 radio that is used in the collar nodes. The measurements were collected by fixing a mobile node in a field near our office and moving another node within a 70m radius around the fixed node. Moving the transmitter in different directions around the receiver was to reduce any biases arising from the antenna radiation pattern not being fully omni-directional. Both nodes remained at an elevation of 1m from the ground. Each point in the figure corresponds to a single RSSI measurement. The solid line represents the maximum distance at which each discrete RSSI value was recorded by the RF212 radio in our measurements. For the purpose of radio ranging, we can use this line as a less conservative estimate of $\hat{r}$ for any RSSI X.

The last bar plot in Figure 17 shows the power consumption and error rate for RSSI-based contact logging. It achieves a similar GPS duty cycle as a fixed contact radius of 5m, but its radio power consumption is higher because it uses the highest transmission power for RSSI ranging. This causes more overhearing at all nodes and increases the radio power consumption. In terms of error rate, the RSSI-based contact logging behaves similarly to plain GPS duty cycling. In summary, RSSI-based ranging has no explicit performance benefits over plain GPS

duty cycling or 5m contact logging, but it provides a more dynamic approach for logging contact at variable distances.

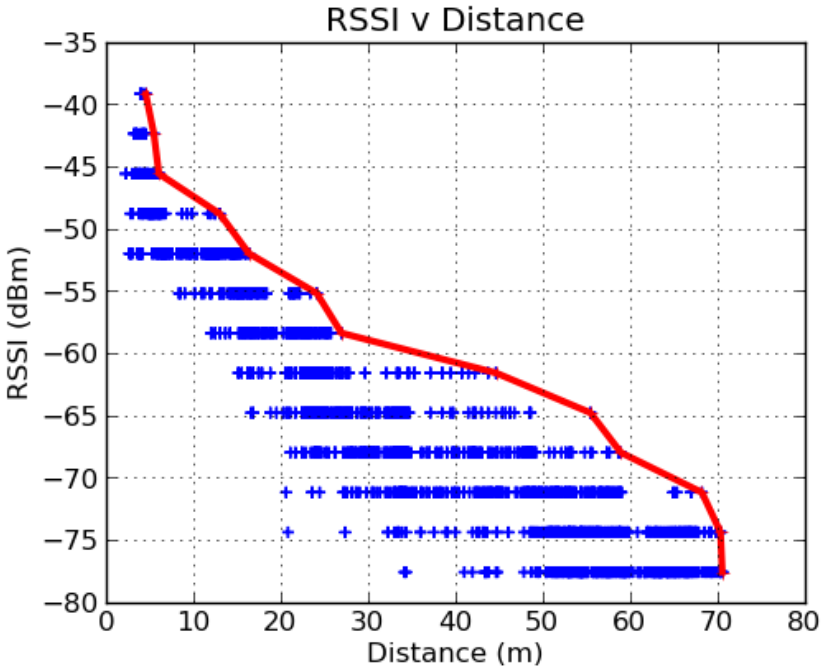


Figure 18: Line-of-Sight RSSI measurements for RF212 Radio. The bounding line represents the maximum possible line-of-sight distance for each observed RSSI value.

5.3 Beaconing Period

The baseline contact logging protocol is periodic, where each node has an internal timer for triggering transmission of contact beacons. Whenever neighbours hear contact beacons, they determine whether to rely on their neighbour's position estimate if that reduces their uncertainty. The choice of contact period is dependent on many independent factors, including the clustering patterns of nodes, their relative speeds, terrain, and AAU. This suggests the need for adapting the contact beacon period to these factors, which is a non-trivial optimisation problem with highly dynamic and unpredictable inputs.

Before attempting to solve this problem, we first explore the impact of the contact beacon period on the accuracy/energy tradeoff in the first part of this section. Next, we propose an event-driven contact beacon transmission, based on local position state changes.

5.3.1 Periodic Beaconing

To investigate the effect of beacon period, Figure 19 shows the power consumption and error rate for contact beacon periods of 1, 5, 10, 20, and 50 seconds. Increasing the contact beacon period from 1 to 5 seconds reduces both power consumption and error rate. While a longer beacon period causes GPS power consumption to increase, it also shrinks the radio power consumption. For beacon periods of 1 second, we observe that the GPS power consumption remains relatively high. The reason is that, on many occasions, a node powers its GPS on to get lock, only to turn it off before getting lock, due to the reception of a useful message from a neighbour. This continuous state switching of the GPS module prevents it from lowering its consumption further with short beacon periods.

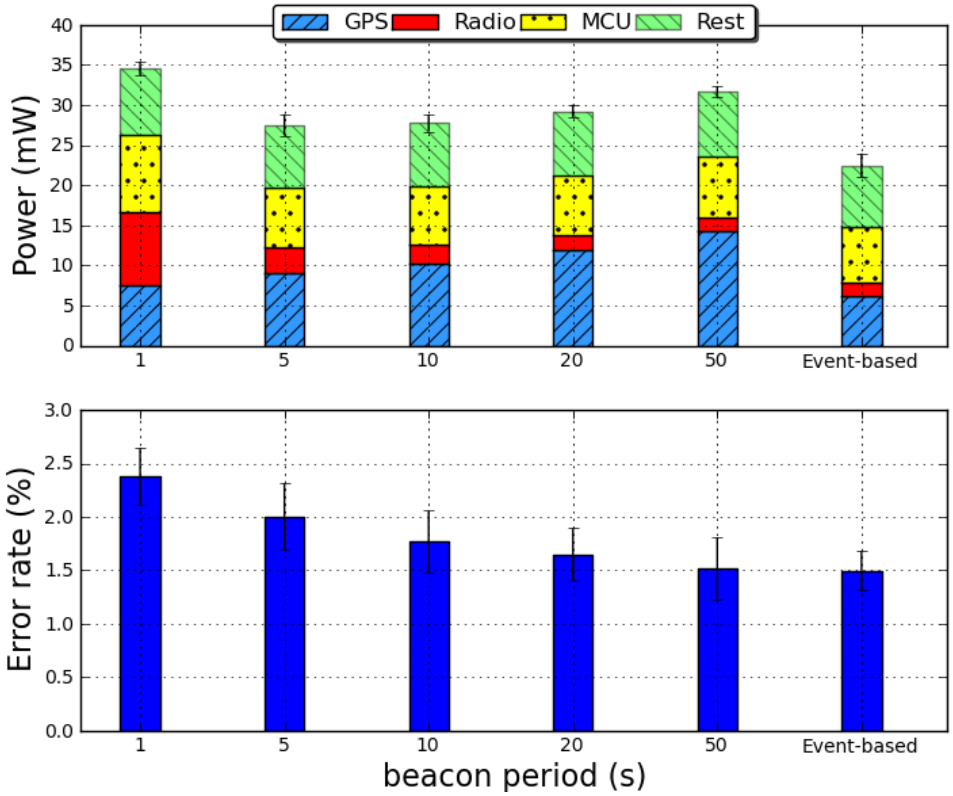


Figure 19: Periodic and event-based beacons

5.3.2 Event-based Beaconing

An additional effect with periodic beaconing is the implicit synchronization of several neighbours of a node that obtains lock. When the node obtains lock, its beacon informs its neighbours of this event, and all neighbours set their uncertainties accordingly. If the neighbours happen to be within the same $\hat{r}$ from the node, and with similar assumed speeds $\overline{s}$, then they are likely to turn on their GPS modules at nearly the same time during the next cycle.

To counter this effect and to reduce radio power consumption further, we propose event-based beacons as an alternative to periodic beaconing. In this approach, a node send its contact beacon whenever it locally detects one of the following position state changes: (1) its position is updated upon obtaining GPS lock; (2) its position is updated upon receiving a useful beacon from one of its neighbours; or (3) it has powered on its GPS module and is in the process of obtaining GPS lock. A beacon sent in state (3) serves as a GPS lock backoff beacon, and its purpose is to inform neighbouring nodes that it will imminently obtain lock, thereby avoiding the implicit synchronization issue.

To support event-based beaconing, nodes use a modified beacon format that includes the expected time to obtain lock (based on history of lock times and off times).This beacon is known as the GPS lock backoff beacon. Figure 20 illustrates how event-based beaconing works. At time t_1, node A obtains a GPS lock. As a result, it sets its uncertainty to $U_{gps}(t_1)$ and broadcasts a beacon declaring its id, uncertainty, and position. Node B receives this beacon at time t_2, and, after determining that contact with node A reduces its own uncertainty, sets its uncertainty according to Equation 4. Nodes A and B then proceed to grow their uncertainty region according to their local speed models. At time t_3, node B hears a beacon from node C with a low uncertainty, and updates its local uncertainty region. Subsequently, it broadcasts a beacon announcing its updated uncertainty, enabling node A to reduce its uncertainty as well at time t_4. By time t_5, the uncertainty at node A grows close to AAU, so A decides to power on its GPS module to avoid exceeding AAU. It also broadcasts a GPS lock backoff beacon informing

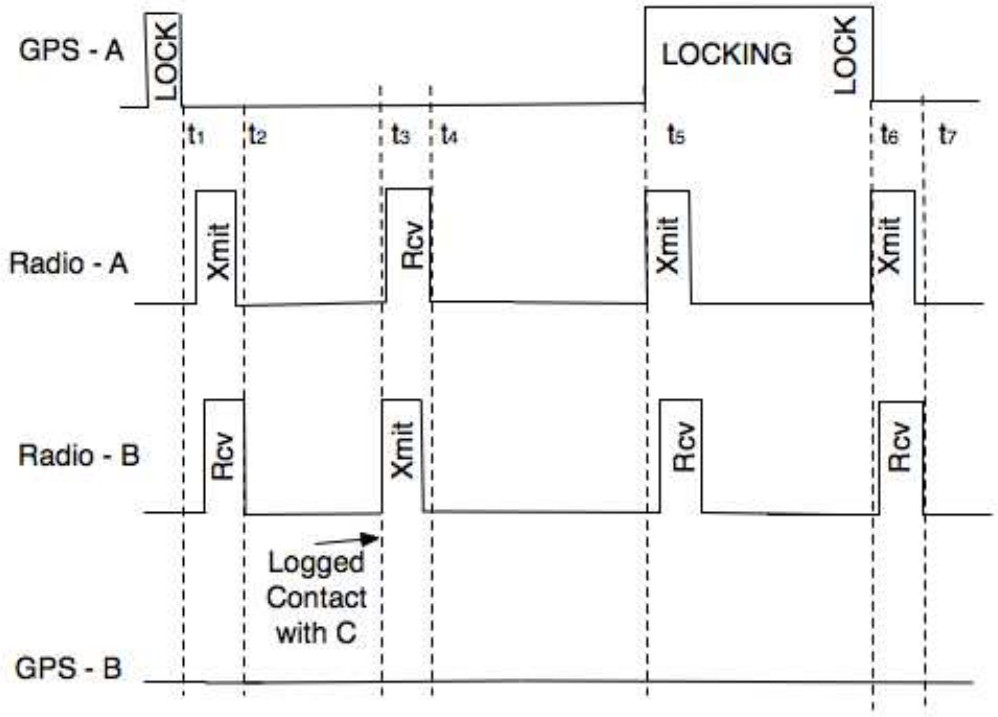


Figure 20: Event-triggered beaconing example: nodes only send beacons when they are getting lock, they get lock, or they log a useful contact with a neighbor

all its neighbours of this action to avoid multiple neighbours trying to simultaneously get GPS lock. Node B hears this beacon, and refrains from powering on its GPS module to get lock. At time t_6, node A succeeds in getting lock, reducing its uncertainty to $U_{gps}(t_6)$, and broadcasts a beacon informing neighbours of this change. Finally, B hears node A's beacon at time t_7 and updates its uncertainty as well. During all these exchanges, four radio packets are transmitted and received, node A's GPS obtained 2 locks, while node B's GPS module remained in off mode.

The last bar plot of of Figure 19 shows the results for event-based beaconing. It achieves both the lowest power consumption and error rate for all of the configurations we consider. The energy saving from event-based beaconing stems from 2 factors: (1) the reduced radio beaconing enables the radio to sleep more often; and (2) the reduced overlaps in GPS lock attempts significantly reduce the GPS power consumption.

Finally, we consider the effect of combining RSSI-based contact logging and event-based beaconing on error rate and power consumption. Figure 21 summarises the results of the various contact radius and beacon configurations in the paper, including GPS duty cycling, GPS duty cycling with contact logging (contact radius of 5m and beacon period of 5 seconds), RSSI-based contact logging with 5 second beacon period, event-based beaconing with a 5 m contact radius, and event- and RSSI- based contact logging. The latter yields the lowest error rate at 1% while increasing power consumption to 28mW. The increased power consumption of this strategy stems from the use of the highest transmit power for RSSI-based beaconing and from the fact that RSSI-bounding of uncertainty is rarely lower than 5m (which causes more frequent GPS locks). This means that event-based beaconing with a 5 m contact radius remains the most energy-efficient approach while providing an error rate that is well within the application's error tolerance of 5%.

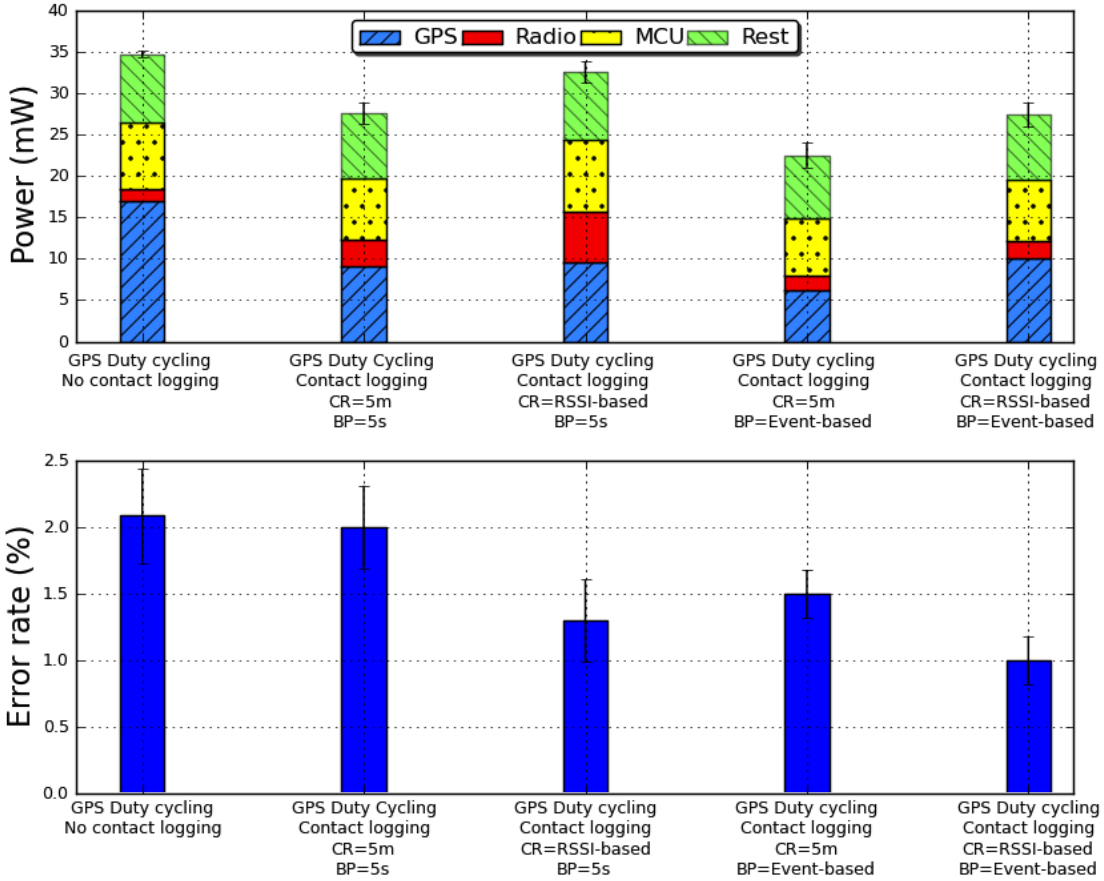


Figure 21: Comparison of configurations. Combination of RSSI-based and event-based beaconing provides the lowest error rate, but the event-based approach with a 5 m contact radius appears to provide the lowest energy consumption while remaining well within the error tolerance.

5.4 Beacon Forwarding

The previous section has shown that event-driven contact beacons can improve both the error rate and the power consumption of mobile nodes. However, the utility and cost of forwarding these contact beacons over multiple hops remains unclear. Multi-hop forwarding of contact beacons can contribute to bounding position uncertainties further to achieve lower GPS duty cycles, but it also leads to more packet transmissions and receptions, which increases energy consumption and can cause congestion.

To investigate this issue, we implement three versions of event-based contact logging in our simulator:

1. No forwarding: nodes that receive a useful contact beacon do not forward it.
2. One-hop forwarding: a useful beacon is only forwarded once.
3. Multi-hop forwarding: there is no hard limit on the forwarding of beacons. Nodes forward a beacon as long as they deem it useful.

Forwarding strategy (1) limits the number of beacons transmitted and received but also constrains the utility of a beacon to a single radio neighbourhood around a node that acquires GPS lock. Strategies (2) and (3) increase the utility region around the node with GPS lock, but they also risk creating implicit synchronisation between nodes that receive a useful beacon. This may cause several nodes to try to forward the beacon at the same time instant, leading to collisions,

Strategy	Unc. Gain (m)	Beacons Tx	Beacons Rx	Useful Beacons	e	θ
None	715088	11099	51095	37476	64.43	0.733
1-hop	789861	31227	223012	41190	25.29	0.185
Multi-hop	829164	49892	370458	46386	16.62	0.125

Table 3: Dynamics of the 3 forwarding strategies

retransmissions, and packet delays. We implement a simple back-off mechanism to mitigate any implicit synchronisation, where a node receiving a useful beacon backs off for a random time within a maximum time window prior to forwarding the beacon. We set a nominal time window of 1 second.

Using our simulator, we evaluate the impact of the 3 forwarding strategies, observing the number of transmitted, received, and useful beacons for each forwarding strategy. We also track the total uncertainty gain in meters for each strategy, which is the aggregated difference in distance between a node's old and new positions upon receiving a useful beacon. We run the simulation over 1 day for the whole cow position dataset with a fixed contact radius of 5m and event-based beaconing. We also define a beacon utility function:

$$e = \frac{uncertainty\ gain}{transmitted\ beacons}$$

to quantify the average uncertainty gain per transmitted beacon and a useful beacon rate θ as the ratio of useful beacons to received beacons. Table 3 summarises the results. Multi-hop forwarding has an incremental improvement in the number of useful beacons and uncertainty gain, but that comes at the cost of nearly 50% increase in transmitted and received beacons. These figures deliver a higher useful rate and beacon utility for the no forwarding case compared to both 1-hop forwarding and multi-hop forwarding.

Figure 22 sheds further light on these results. The strategies consume similar energy and yield similar error rates, with a 10% improvement for the multi-hop forwarding strategy for both those metrics. The tradeoff in energy is clear: the no forwarding strategy has the lowest radio power consumption, but its GPS has to turn on more often; the multi-hop forwarding strategy is overly aggressive in forwarding beacons, resulting in higher radio power consumption yet lower GPS power consumption. Note that while the multi-hop forwarding strategy does not yield significant energy savings for our scenario, its benefits may increase for other applications with different mobility patterns. The metrics we have presented in this section can be used to determine the utility of multi-hop beacon forwarding for new applications.

5.5 Summary

This section has investigated several approaches for combining radio ranging with GPS duty cycling to save energy in outdoor localisation systems. Event-based beaconing with a constant contact radius of 5m performs best for our application. While beacon forwarding provides incremental energy savings, single hop event-based beaconing is simpler to implement and manage, and it remains our strategy of choice.

6 Conclusion and Perspectives

This paper has proposed and validated short-range radio contact logging as an effective complement to GPS duty cycling for balancing the positioning accuracy and energy-efficiency tradeoff. Given application constraints that comprise a target lifetime, acceptable uncertainty, and error tolerance, we have established a strategy for tailoring the contact radius and beacon period

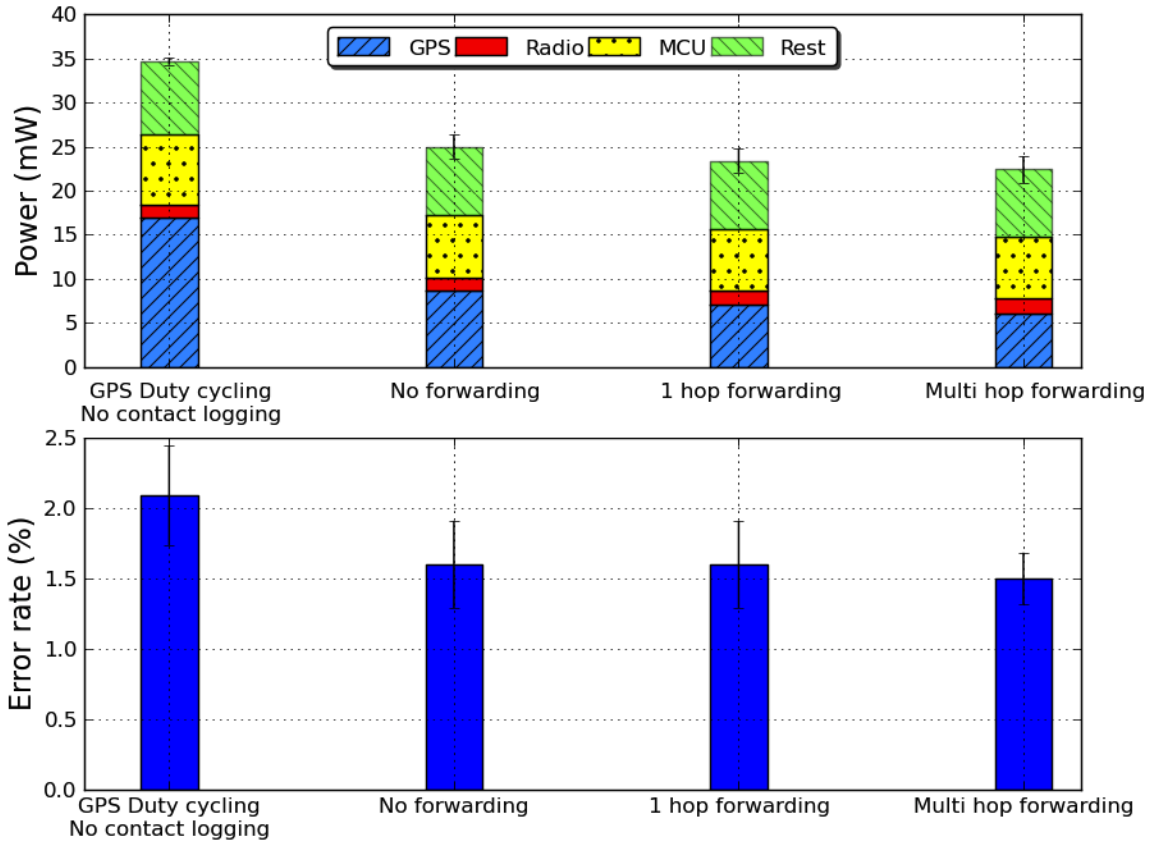


Figure 22: Impact of beacon forwarding strategy on power consumption and error rates: multi-hop beacon forwarding marginally improves performance. All contact logging results use event-based beaconing.

for any mobile sensor network application. We have shown how our strategy can rely on key features of GPS modules, through empirical characterisation of two generations of modules. A key contribution of this paper is to define a target accuracy for the GPS module that minimises the active time of the module within a duty cycling strategy. We have determined through simulations based on our empirical data set that a contact radius of 5m performs best for our application. We then proceeded to identify the most suitable beacon period for our dataset, based on which we proposed event-driven beaconing that reactively sends beacons only when nodes detect state changes.

While the event-driven beaconing strategy yields the lowest power consumption and error rate in our scenario, the optimal strategy is highly dependent on the mobility pattern of the nodes, their clustering characteristics, and application requirements. For instance, an application with highly mobile nodes may benefit from a more proactive beaconing strategy that transmits beacons at short periodic intervals. RSSI-based contact logging did not yield significant benefits over statically set contact radii, except for providing a dynamic contact radius. It may not suit radios with no RSSI or high multi-path deployments.

We have also studied the impact of beacon forwarding. Multi-hop forwarding results in a 10% improvement in both energy consumption and error rate over the no-forwarding strategy. Multi-hop forwarding reduces GPS duty cycles at the cost of increased radio duty cycles. Because our simulations consider a static AAU of 50m, the opportunities for multi-hop forwarding of beacons are limited as beacons stop being useful after 2 or 3 hops. We expect that a dynamic

AAU would benefit more from multi-hop beaconing, since useful beacons can propagate for a larger number of hops when the AAU is higher.

The static nodes that serve as a routing backbone for the mobile nodes could also serve as local positioning anchors. The area of relative positioning through powerful local beacons is indeed a well-investigated area. While our current virtual fencing deployments do include static nodes that can be used for this purpose, we are aiming towards infrastructure-free deployments for high scalability in monitoring mobile devices. The GPS duty cycling and radio contact logging strategies here represent a step towards infrastructure-free deployments. However, deployments that do include static nodes for operational purposes can certainly enable mobile nodes to perform radio triangulation when they are in range of one or more static nodes.

Another possible enabler of more sparse deployments is the use of inertial sensors, such as accelerometers and magnetometers, to refine position estimates while the GPS is powered off. While this is a proven strategy in other GPS tracking scenarios, its utility for low power devices is an open question. The accelerometer in particular involves an inherent tradeoff between its accuracy for predicting speed and its sampling rate. It can only provide precise estimates of speed for high sample rates, which incurs high energy cost. The use of accelerometers as boolean indicators of motion is certainly an option for input to the algorithm. Magnetometers can certainly provide accurate heading data for low energy cost, thereby limiting the uncertainty growth within a angular cone rather than a circle. This comes at a cost of increasingly computational complexity for growing the irregular uncertainty region as the mobile nodes change directions. Finally, the use of pedometers can be useful for counting footsteps of animals or people, and can serve as an additional input to the speed and distance estimation.

Some concepts from this paper, such as dynamically changing the AAU according to distance from the fence, are specific to the cattle monitoring application. However, our overall strategy is applicable to other mobile tracking applications, such as smart phone social networking applications. Consider groups of friends with smart phones that regularly meet and use location-based services that rely on GPS. If all smart phones keep their GPS modules on to run the services, they will most likely deplete their batteries quickly. Alternatively, the use of contact logging to share the GPS load among the smart phones can prolong the lifetime of all devices. Applying our strategy to this application would require mobility models, clustering patterns, and application-specific user policies to determine the best configuration.

Acknowledgements

This work was supported by the Sensors and Sensor Networks Transformation Capability Platform at CSIRO. The authors would like to thank Greg Bishop-Hurley, Philip Valencia, and Brano Kusy for their valuable inputs in realising this work.

References

[BCPR04] Z. Butler, P. Corke, R. Peterson, and D. Rus. Virtual fences for controlling cows. In *Proc. Int. Conf. Robotics and Automation*, pages 4429–4436, New Orleans, April 2004.

[CCC+06] Li-Wei Chan, Ji-Rung Chiang, Yi-Chao Chen, Chia nan Ke, Jane Yung jen Hsu, and Hao-Hua Chu. Collaborative localization: Enhancing wifi-based position estimation with neighborhood links in clusters. In *Pervasive*, pages 50–66, 2006.

[CGea09] I. Constandache, S. Gaonkar, and et al. Enloc: Energy-efficient localization for mobile phones. In *INFOCOM*, pages 2716–2720, 2009.

[GPC+09] Y. Guo, G. Poulton, P. Corke, G.J. Bishop-Hurley, T. Wark, and D.L. Swain. Using accelerometer, high sample rate gps and magnetometer data to develop a cattle movement and behaviour model. *Ecological Modelling*, 220(17):2068 – 2075, 2009.

[ID03] M. Ilyas and R. C. Dorf, editors. *The handbook of ad hoc wireless networks*. CRC Press, Inc., Boca Raton, FL, USA, 2003.

[JCDS10] Raja Jurdak, Peter Corke, Dhinesh Dharman, and Guillaume Salagnac. Adaptive gps duty cycling and radio ranging for energy-efficient localization. In *ACM Conference on Embedded Networked Sensor Systems (Sensys)*, pages 57–70, Switzerland 2010.

[JRO08] R. Jurdak, A. G. Ruzzelli, and G.M. P. O'Hare. Adaptive radio modes in sensor networks: How deep to sleep? In *IEEE (SECON)*, pages 386–394, June 2008.

[KLGT09] Mikkel Baun Kjaergaard, Jakob Langdal, Torben Godsk, and Thomas Toftkjaer. Entracked: energy-efficient robust position tracking for mobile devices. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, pages 221–234, New York, NY, USA, 2009. ACM.

[KSB+07] Branislav Kusy, Janos Sallai, Gyorgy Balogh, Akos Ledeczi, Vladimir Protopopescu, Johnny Tolliver, Frank DeNap, and Morey Parang. Radio interferometric tracking of mobile wireless nodes. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, MobiSys '07, pages 139–151, New York, NY, USA, 2007. ACM.

[LKea10] K. Lin, A. Kansal, and et al. Energy-accuracy trade-off for continuous mobile device location. In *MobiSys*, pages 285–298, 2010.

[LNR04] G. Lin, G. Noubir, and R. Rajaraman. Mobility models for ad hoc network simulation. In *INFOCOM*, 2004.

[LZZ06] J. Liu, Y. Zhang, and F. Zhao. Robust distributed node localization with error management. In *MobiHoc*, pages 250–261, 2006.

[PKG10] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *MobiSys*, pages 299–314, 2010.

[PPK03] S. Pattern, S. Poduri, and B. Krishnamachari. Energy-quality tradeoffs in sensor track- ing: Selective activation with noisy measurements. In *IPSN*, 2003.

[PPP11] B. Patil, R. Patil, and A. Pittet. Energy saving techniques for gps based tracking applications. In *Integrated Communications, Navigation and Surveilance Conference (ICNS), 2011*, pages J8–1 –J8–10, may 2011.

[SL06] K. Srinivasan and P. Levis. Rssi is under appreciated. In *EmNets*, 2006.

[TKAGK05] Sameer Tilak, Vinay Kolar, Nael B. Abu-Ghazaleh, and Kyoung-Don Kang. Dynamic localization control for mobile sensor networks. In *IPCCC*, pages 587–592, 2005.

[TSBW04] Bjørn Thorstensen, Tore Syversen, Trond-Are Bjørnvold, and Tron Walseth. Electronic shepherd - a low-cost, low-bandwidth, wireless network system. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, MobiSys '04, pages 245–255, New York, NY, USA, 2004. ACM.

[uba] u blox. TIM-4x Data Sheet. http://www.u-blox.com/images/downloads/Product_Docs/TIM-4x_Data_Sheet%28GPS.G4-MS4-07013%29.pdf.

[ubb] u blox. UBX-G5010 Product Summary. http://www.u-blox.com/images/downloads/Product_Docs/UBX-G5010_Prod_Summary%28GPS.G5-X-06042%29.pdf.

[WCH+07] Tim Wark, Chris Crossman, Wen Hu, Ying Guo, Philip Valencia, Pavan Sikka, Peter Corke, Caroline Lee, John Henshall, Kishore Prayaga, Julian O'Grady, Matt Reed, and Andrew Fisher. The design and evaluation of a mobile sensor/actuator network for autonomous animal control. In *Proceedings of the 6th international conference on Information processing in sensor networks*, IPSN '07, pages 206–215, New York, NY, USA, 2007. ACM.

[WXL04] J. Winter, Y. Xu, and W.C. Lee. Prediction-based strategies for energy saving in object tracking sensor networks. In *IEEE ICMDM*, January 2004.

[YHC+08] Chuang-Wen You, Polly Huang, Hao-hua Chu, Yi-Chao Chen, Ji-Rung Chiang, and Seng-Yong Lau. Impact of sensor-enhanced mobility prediction on the design of energy-efficient localization. *Ad Hoc Netw.*, 6:1221–1237, November 2008.

[You08] C.-W. You. *Enabling Energy-Efficient Localization Services on Sensor Network Positioning Systems*. PhD thesis, National Taiwan University, January 2008.

[Zea05] P. Zhang and C. Sadler et al. *Habitat Monitoring with ZebraNet: Design and Experiences*, chapter Wireless Sensor Networks: A Systems Perspective. Artech House, 2005.